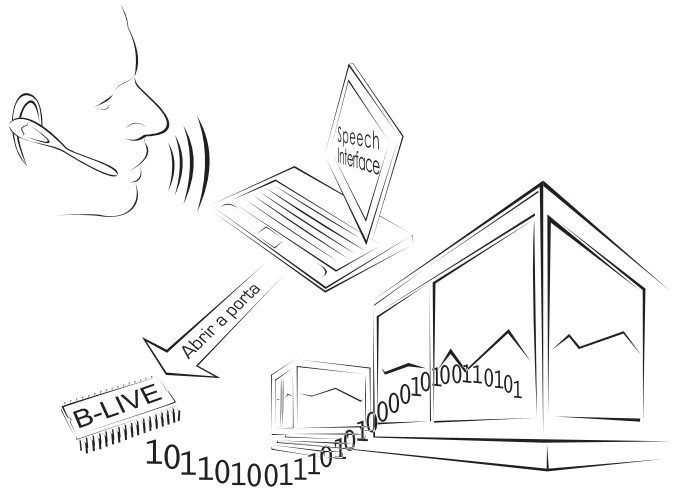
Departamento de
Universidade de Aveiro Electrónica, Telecomunicações e Informática.
2007

CARLOS JORGE ENES CAPITÃO DE ABREU

INTERFACE COM RECONHECIMENTO DE FALA PARA APOIO A PESSOAS COM LIMITAÇÕES FUNCIONAIS



CARLOS JORGE ENES CAPITÃO DE ABREU

INTERFACE COM RECONHECIMENTO DE FALA PARA APOIO A PESSOAS COM LIMITAÇÕES FUNCIONAIS

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Biomédica - Ramo Instrumentação, Sinal e Imagem Médica, realizada sob a orientação científica do Doutor José Alberto Gouveia Fonseca, Professor Associado e do Doutor António Joaquim da Silva Teixeira, Professor Auxiliar, ambos do Departamento de Electrónica, Telecomunicações e Informática da Universidade de Aveiro.

Dedico este trabalho a toda a minha família, em especial à minha Irene.

O fruto de cada palavra retorna a quem a pronunciou.

Abu Shakur

O júri

Presidente

Doutora Beatriz Sousa Santos
Professora Associada com Agregação da Universidade de Aveiro.

Vogais

Doutor José Alberto Gouveia Fonseca
Professor Associado da Universidade de Aveiro (Orientador).

Doutor António Joaquim da Silva Teixeira
Professor Auxiliar da Universidade de Aveiro (Co-Orientador).

Doutor Francisco Godinho
Professor Auxiliar Convidado da Universidade de Trás-os-Montes e Alto Douro.

Agradecimentos

Aproveito esta oportunidade para agradecer a todos os Professores que, ao longo dos últimos anos, partilharam comigo os seus conhecimentos e, dessa forma, permitiram que atingisse esta fase da minha formação académica.

Aos Professores António Joaquim da Silva Teixeira e José Alberto Gouveia Fonseca, um agradecimento especial pela sua orientação durante a elaboração deste trabalho.

À Micro I/O pelos meios disponibilizados, sem os quais, não teria sido possível realizar este trabalho.

Ao Centro de Medicina de Reabilitação da Região Centro - Rovisco Pais, pela cooperação disponibilização dos meios necessários à realização dos ensaios com utilizadores em recuperação.

À Dr.ª Arminda Lopes, pela disponibilidade e acompanhamento aos doentes durante os ensaios.

À Manuela, ao Márcio e ao joão Eduardo, pela disponibilidade e cooperação durante a realização dos ensaios.

Ao Dr. Emílio Enes pela revisão atenta que fez a estas páginas.

Ao *Microsoft Language Development Center*, por disponibilizar o reconhecedor de fala multi-utilizador, sem o qual não teria sido possível obter resultados tão animadores.

Palavras-chave

Interfaces Humano-máquina (HMI), domótica, limitações funcionais, reconhecimento de fala.

Resumo

A utilização de tecnologia nas habitações domésticas é uma realidade crescente. Os sistemas domóticos apoiam-nos na realização de um sem número de tarefas quotidianas. Além do conforto que proporcionam, estes sistemas permitem que as pessoas com limitações funcionais tenham maior autonomia e mobilidade dentro das suas habitações. Existem, contudo, alguns casos em que as interfaces entre os sistemas domóticos e as pessoas com limitações funcionais não são as mais adequadas. O objectivo deste trabalho é desenvolver uma interface de fácil utilização entre pessoas com limitações funcionais e o sistema domótico B-LIVE. Tendo em conta as limitações físicas destas pessoas, a fala, como interface humano-máquina, foi o ponto de partida para o trabalho desenvolvido. Esta dissertação apresenta uma interface humano-máquina com reconhecimento de fala, tendo em vista a sua utilização por pessoas com limitações funcionais para que estas possam interagir com o B-LIVE. Foram desenvolvidas duas versões da interface, uma com um reconhecedor dependente do orador e outra com um reconhecedor independente do orador. Em ambos os casos os resultados obtidos (quer em ambiente laboratorial, quer em utilização real), permitem concluir que a fala é uma interface viável para a utilização em questão.

Keywords Human-machine interfaces (HMI), home automation, disabled people, speech recognition.

Abstract The utilization of technology in our homes is a growing reality. Home automation can support us in many of our daily tasks. Apart from the comfort that these systems provide, they allow disabled people to achieve more autonomy and mobility within their homes. There are, however, some cases where the interfaces between home automation systems and disabled people are not the most appropriate. The purpose of this work is to develop a user friendly interface between disabled people and the home automation system B-LIVE. Given the physical limitations of these people, speech as human-machine interface was the starting point for this work. This dissertation presents a human-machine interface with speech recognition, that can be used by disabled people so that they can interact with the B-LIVE system. We have developed two versions of the interface: one with a speaker dependent speech recogniser and another with a speaker independent speech recogniser. In both cases the results (both in laboratory environment and in real utilization), suggest that speech is a viable interface to be used in these applications.

Conteúdo

1 Introdução **1**

1.1 Necessidades Especiais . . . 2
1.2 Enquadramento . . . 2
1.3 A fala como interface humano-máquina . . . 3
1.4 Objectivos . . . 4
1.5 Organização da presente dissertação . . . 4
1.6 Resultados já publicados . . . 6

2 A fala como meio de comunicação entre Humanos **7**

2.1 Produção de sinais acústicos de fala . . . 8
2.1.1 Constituição do aparelho produtor humano . . . 10
2.1.2 Princípios de funcionamento do aparelho produtor humano . . . 12
2.1.3 Classificação dos sons da fala . . . 13
2.2 Percepção de fala . . . 16
2.2.1 Constituição do aparelho auditivo humano . . . 18
2.2.2 Princípio de funcionamento do aparelho auditivo humano . . . 19
2.3 Coordenação e controlo dos aparelhos produtor e auditivo . . . 20
2.3.1 Enervação do aparelho produtor humano . . . 23
2.3.2 Enervação do aparelho auditivo humano . . . 23
2.4 Comentários finais . . . 23

3 Reconhecimento de fala **25**

3.1 Definição do problema . . . 26
3.2 Componentes de um reconhecedor típico . . . 29
3.3 Extracção de parâmetros . . . 31
3.3.1 Pré-ênfase . . . 31
3.3.2 Aplicação da janela de análise . . . 31
3.3.3 Coeficientes *Mel Frequency Cepstral Coefficients* . . . 32
3.3.4 Energia e coeficientes delta . . . 33
3.4 Modelo da linguagem . . . 34
3.4.1 Modelos N-grams . . . 34
3.4.2 Modelos *Finite State Model* . . . 35
3.4.3 Perplexidade . . . 35
3.5 Modelos Acústicos . . . 36
3.5.1 Modelos de Markov não observáveis . . . 37

3.5.2 Sistemas de reconhecimento da fala baseados em modelos de Markov não observáveis . . . 40
3.5.3 Treino de reconhecedores de fala . . . 41
3.6 Descodificador . . . 42
3.7 Avaliação . . . 43
3.8 Comentários finais . . . 44

4 Desenvolvimento de uma interface *Speech Enabled* para pessoas com limitações funcionais 45
4.1 Princípio de funcionamento . . . 46
4.1.1 Reconhecimento de fala . . . 46
4.1.2 Análise e validação . . . 47
4.1.3 Actuação . . . 48
4.2 Arquitectura da aplicação de interface . . . 49
4.2.1 *Interface Layer* . . . 49
4.2.2 *Business Layer* . . . 50
4.2.3 *Hardware Layer* . . . 52
4.2.4 *Database Layer* . . . 55
4.2.5 *External Connection Layer* . . . 57
4.2.6 *Recognizer Layer* . . . 58
4.3 Base de Dados . . . 59
4.4 Sistema domótico para pessoas com limitações funcionais: B-LIVE . . . 61
4.4.1 Arquitectura do sistema B-LIVE . . . 62
4.4.2 Protocolo de comunicações utilizado pelo B-LIVE . . . 63
4.4.3 *Firmware* B-LIVE . . . 64
4.4.4 Arquitectura do *Firmware* B-LIVE . . . 65
4.4.5 Operação do sistema B-LIVE . . . 66
4.4.6 Conclusão . . . 67
4.5 Selecção da ferramenta de reconhecimento de fala, para construir o reconhecedor dependente do orador . . . 67
4.5.1 *Hidden Markov Model Toolkit* . . . 69
4.5.2 Projecto *Sphinx* . . . 72
4.5.3 Conclusão . . . 75
4.6 Reconhecedor de fala dependente do orador, baseado em HTK . . . 76
4.6.1 Configurações e preparação dos dados . . . 77
4.6.2 Criação dos modelos monofones . . . 82
4.6.3 Criação dos modelos trifones . . . 84
4.6.4 Avaliação do reconhecedor . . . 84
4.6.5 Reconhecimento em tempo real . . . 85
4.7 Reconhecedor de fala independente do orador . . . 85
4.8 Comentários finais . . . 89

5 Resultados 91
5.1 Avaliação dos resultados obtidos com o reconhecedor dependente do orador . . . 91
5.1.1 Reconhecimento com modelos monofones . . . 92
5.1.2 Reconhecimento com modelos trifones . . . 92
5.1.3 Reconhecimento em tempo real . . . 93

5.2 Avaliação dos resultados obtidos com o reconhecedor independente do orador . . . 94
5.3 Avaliação dos resultados obtidos em utilização real no CMRRC-Rovisco Pais 95
5.4 Comentários finais . . . 98

6 Conclusões **99**
6.1 Resumo do Trabalho . . . 99
6.1.1 Tecnologias disponíveis no mercado . . . 100
6.1.2 Escolha da ferramenta de reconhecimento a utilizar . . . 100
6.1.3 Bases teóricas sobre reconhecimento de fala . . . 101
6.1.4 Limitações das pessoas com tetra e paraplegia em produzir sons de fala . . . 101
6.1.5 Desenvolvimento da aplicação de interface . . . 101
6.1.6 Desenvolvimento dos reconhecedores de fala . . . 102
6.1.7 Avaliação da interface . . . 102
6.2 Principais Resultados . . . 103
6.3 Sugestões para Continuação . . . 104

A Alfabetos fonéticos **105**

B Reconhecedor de fala dependente do orador, baseado em HTK **109**
B.1 Configurações e preparação dos dados . . . 109
B.1.1 Gramática . . . 110
B.1.2 Criação dos conjuntos de frases para treino e teste . . . 113
B.1.3 Dicionário . . . 113
B.1.4 Gravação das frases para treino e teste . . . 114
B.1.5 Criação dos ficheiros com a transcrição fonética . . . 116
B.1.6 Extracção dos *feature vectors* . . . 117
B.2 Criação dos modelos monofones . . . 119
B.2.1 Inicialização dos modelos monofones . . . 119
B.2.2 Ajuste do modelo de silêncio e introdução de pausas curtas . . . 121
B.2.3 Realinhamento dos dados . . . 122
B.3 Criação dos modelos trifones . . . 123
B.3.1 Criação dos modelos trifones a partir dos monofones . . . 123
B.4 Avaliação do reconhecedor . . . 125
B.4.1 Avaliação dos monofones . . . 126
B.4.2 Avaliação dos trifones . . . 126
B.5 Reconhecimento em tempo real . . . 126

C Avaliação e questionário **127**
C.1 Cenário 1: O utilizador encontra-se na sala e pretende ir à casa de banho. . . . 127
C.2 Cenário 2: O utilizador está no quarto e quer sair para a rua. . . . 128
C.3 Questionário . . . 129
C.4 Respostas ao questionário . . . 131
C.4.1 Utilizador M . . . 131
C.4.2 Utilizador F . . . 132

D Termo de consentimento informado **135**

Lista de Figuras

1.1 População sem e com deficiência em Portugal, Censos 2001 [7]. 2

2.1 Cadeia da fala: produção, transmissão e recepção de sinais de fala (adaptado de várias fontes). 8
2.2 Produção da fala. Depois de formular uma mensagem o cérebro envia os sinais nervosos adequados para que os orgãos da fala produzam os respectivos sons (adaptado de várias fontes). 9
2.3 Aparelho produtor humano (adaptado de MIT OCW). 10
2.4 Percepção da fala. Os sons são captados e transformados em impulsos nervosos pelo ouvido. De seguida, os impulsos nervosos são analisados e interpretados pelo cérebro com o objectivo de descodificar a mensagem recebida (adaptado de várias fontes). 17
2.5 Constituição do ouvido humano: ouvido externo (pavilhão auricular, canal auditivo e tímpano); ouvido médio (bigorna, estribo, martelo e a trompa de Eustáquio); ouvido interno (cóclea, vestíbulo, canais semicirculares e nervo auditivo) [21]. 17
2.6 Neurónio: célula principal, dendrites, axónio e sinapses 20
2.7 Nervos cranianos. I-Nervo Olfactivo, II-Nervo Óptico, III-Nervo Oculomotor, IV-Nervo Troclear, V-Nervo Trigémio, VI-Nervo Abducente, VII-Nervo Facial, VIII-Nervo Auditivo, IX-Nervo Glossofaríngeo, X-Nervo Vago, XI-Nervo Acessório, XII-Nervo Hipoglosso [18] [21] [24]. 22

3.1 Diagrama típico de um sistema de reconhecimento de fala [30] 30
3.2 Processamento do sinal de áudio. 31
3.3 Extracção dos vectores de parâmetros do sinal de fala. 32
3.4 Extracção dos coeficientes MFCC. 33
3.5 Gramática do tipo FSM. 35
3.6 HMM ergódico. 39
3.7 HMM do tipo *left-to-rigt*. 39
3.8 HMM de um fonema. 40
3.9 Cálculo da *minimum edit distance* entre duas frases. 44

4.1 Princípio de funcionamento da interface com reconhecimento de fala. 46
4.2 Estrutura das frases. 47
4.3 Correspondência entre os conjuntos de frases reconhecidas e de instruções. 47
4.4 Análise da informação contida nas frases. 48
4.5 Arquitectura da interface *Speech Enabled*. 49
4.6 Interface com o utilizador. Disponibiliza botões para ligar e desligar o reconhecedor e para sair da aplicação. Apresenta a informação sobre o reconhecimento, vinda do reconhecedor, e o comando correspondente. 50

4.7 Diagrama de classes da *Business Layer*. A *Business Layer* utiliza a classe *mioBLiveComm* para aceder às comunicações, a classe *mioJHVite* para aceder ao reconhecedor de fala, as classes *mioRecognitionAnalyser* e *mioCommandInfo* para analisar as sequências de palavras vindas do reconhecedor e as classes *Log* e *Comandos* para aceder às respectivas tabelas na base de dados. 51
4.8 Diagrama de classes da *Hardware Layer*. O acesso à porta série é feito pela classe *mioSerialPort*, a classe *mioSerialFrame* é utilizada para detectar e enviar tramas pela porta série, por fim, a classe *mioBLiveComm* é utilizada para enviar comandos. 53
4.9 Diagrama de classes da Database Layer. A classe *mioAccess* herda a classe *mioEngineBase*. As restantes classes utilizam a *mioEngineBase* para acederem à base de dados. 56
4.10 Diagrama de classes da *External Connection Layer*. 57
4.11 Diagrama de classes da *Recognizer Layer*. 58
4.12 Relações entre as tabelas da base de dados. 61
4.13 Arquitectura do sistema B-LIVE. 62
4.14 Interacção do sistema B-LIVE com o exterior. 62
4.15 Arquitectura dos módulos B-LIVE. 63
4.16 Arquitectura interna dos módulos B-LIVE. 63
4.17 Estrutura das *frames* trocadas pela linha série. 64
4.18 Arquitectura do *firmware* B-LIVE [8]. 65
4.19 Interface interruptor e rato de boca. 67
4.20 Arquitectura da *Hidden Markov Model Toolkit* [40]. 70
4.21 Arquitectura do *Sphinx-4* [41]. 73
4.22 Construção do reconhecedor. 77
4.23 Cenário de utilização do reconhecedor. 78
4.24 Gramática. 80
4.25 Frases iniciadas por “Ligar”e “Desligar”. 80
4.26 Protótipo para os HMMs. 83
4.27 Arquitectura da Microsoft Speech API 5.3. Múltiplas aplicações podem partilhar os *speech engines* disponíveis através da SAPI *Runtime*. 85
4.28 Estrutura da gramática utilizada pela SAPI 5.3. 88
4.29 Gramática utilizada pelo reconhecedor independente do orador. 88

5.1 Relatório da avaliação feita aos modelos monofones. 92
5.2 Relatório da avaliação feita aos modelos trifones. 93
5.3 Resultados obtidos com o reconhecedor dependente do orador no reconhecimento em tempo real. 94
5.4 Dados sobre a naturalidade, sexo e idade dos utilizadores que avaliaram o desempenho do reconhecedor independente do orador. 95
5.5 Resultados obtidos no reconhecimento em tempo real, com o reconhecedor independente do orador. 96

A.1 Alfabeto fonético internacional [50]. 106
A.2 Alfabeto fonético SAMPA [23]. 107

B.1 Construção do reconhecedor. 109
B.2 Cenário de utilização do reconhecedor. 110
B.3 Gramática. 112

B.4 Frases iniciadas por “Ligar”e “Desligar”. 112
B.5 Conjunto de treino. 113
B.6 Dicionário. 115
B.7 Lista de fonemas utilizados. 115
B.8 Aplicação para gravação das frases de treino e teste. 116
B.9 Transcrições ao nível da palavra. 117
B.10 Configurações para gerar a transcrição fonética. 117
B.11 Transcrições com monofones. 118
B.12 Ficheiro de configuração. 119
B.13 Correspondência entre os ficheiros de dados e de *feature vectors*. 119
B.14 Protótipo para os HMMs. 120
B.15 Comandos para ajustar o modelo de silêncio. 121
B.16 Modelo para as pausas curtas. 122
B.17 Introdução das pausas curtas. 123
B.18 Transcrições com trifones. 124
B.19 Lista de trifones. 125
B.20 Configuração utilizada para gerar os modelos trifones. 125

Lista de Tabelas

2.1 Partes do aparelho produtor Humano: Produtores, Vibrador, Ressoadores, Articuladores e Sensor/Coordenador. 11
2.2 Consoantes da língua portuguesa e sua classificação, quanto ao modo de articulação. Representadas segundo a nomenclatura SAMPA [23]. 14
2.3 Vogais orais da língua portuguesa e sua classificação [20]. Representadas segundo a nomenclatura SAMPA [23]. 16
2.4 Vogais nasais da língua portuguesa e sua classificação [20]. Representadas segundo a nomenclatura SAMPA [23]. 16
2.5 Classificação dos nervos cranianos e respectivas funções [21]. 21

3.1 Perplexidades típicas para diferentes domínios. 36

C.1 Conjunto de frases para avaliação da interface de reconhecimento de fala. 128

Lista de abreviaturas

ANSI	*American National Standards Institute*
ASCII	*American Standard Code for Information Interchange*
ATIS	*Air Travel Information System*
CAN	*Controller Area Network*
CFG	*Context-Free Grammar*
CMRRC	Centro de Medicina de Reabilitação da Região Centro
CMU	*Carnegie Mellon University*
CU	*Cambridge University*
CUED	*Speech Vision and Robotics Group of the Cambridge University Engineering Department*
DARPA	*Defense Advanced Research Projects Agency*
FFT	*Fast Fourier Transform*
FSM	*Finite State Model*
HMI	*Human-Machine Interface*
HMM	*Hidden Markov Model*
HP	*Hewlett Packard*
HSH	*Health Smart Homes*
HTK	*Hidden Markov Model Toolkit*
IBM	*International Business Machines Corporation*
IDFT	*Inverse Discrete Fourier Transform*
I2C	*Inter-Intergrated Circuit*
INE	Instituto Nacional de Estatística
IPA	*Internatoinal Phonetic Alphabet*

JDBC	*Java Database Connectivity*
LCOMDRV	*Local Communication Driver*
MERL	*Mitsubishi Electric Research Laboratories*
MFCC	*Mel Frequency Cepstral Coefficients*
MIT	*Massachusetts Institute of Technology*
ML	*Maximum Likelihood*
MLDC	*Microsoft Language Development Center*
MLF	*Master Label File*
MMF	*Master Macro File*
MS	*Microsoft*
PC	*Personal Computer*
*RS*232	*Recommended Standard 232*
RCOMDRV	*Remote Communication Driver*
SAMPA	*Speech Assessment Methods Phonetic Alphabet*
SAPI	*Microsoft Speech API*
SLF	*HTK Standard Lattice Format*
SML	*Sun Microsystems Laboratories*
SNC	Sistema Nervoso Central
SNP	Sistema Nervoso Periférico
SPI	*Serial Peripheral Interface*
SQL	*Structured Query Language*
TI	*Texas Instruments*
UCSC	*University of California at Santa Cruz*
WER	*Word Error Rate*
XML	*Extensible Markup Language*

Capítulo 1

Introdução

O desenvolvimento tecnológico das últimas décadas não tem precedentes na história da humanidade. Como consequência, o nível de vida da maioria das pessoas dos países mais desenvolvidos tem aumentado consideravelmente. Este aumento do nível de vida faz com que as pessoas procurem produtos que aumentem o grau de conforto dos locais onde passam a maior parte do seu tempo. Os sistemas automáticos para **controlo de habitações** são disso um bom exemplo.

O conceito de ***Smart Houses*** [1] [2], surgiu da investigação em ***Home Automation* e *Home Networking Areas*** [3]. Este conceito aliado à automação doméstica é bastante útil em aplicações de ***Assistive Technologies***. A utilização de *Assistive Technologies*, em conjunto com *Smart Houses*, deu origem ao conceito de ***Health Smart Homes*** (HSH) [4] [5]. Estas tecnologias não representam apenas uma melhoria de conforto e qualidade de vida, mas também uma nova oportunidade para as pessoas com graves **limitações funcionais**.

No caso particular de pessoas com limitações funcionais, estas tecnologias podem ser utilizadas de diversas formas, por exemplo: na monitorização da evolução dos tratamentos de pessoas em recuperação de acidentes graves, proporcionando mais **independência e integração** às pessoas **tetra ou paraplégicas** e também no auxílio a pessoas **idosas** na realização de tarefas quotidianas.

1.1 Necessidades Especiais

Segundo os dados obtidos através do Instituto Nacional de Estatística (INE), tendo em conta os **Censos de 2001**, existem em Portugal continental cerca de $1.693.493$ **habitantes com idade igual ou superior a** 65 **anos e** 634.408 **com algum tipo de deficiência** [6] [7]. A partir destes dados facilmente se verifica que são muitos os potenciais utilizadores de sistemas de HSH.

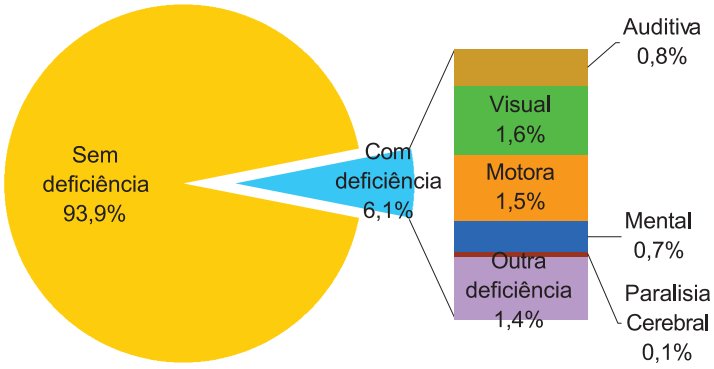


Figura 1.1: População sem e com deficiência em Portugal, Censos 2001 [7].

Os dados do INE na figura 1.1, mostram que uma grande fatia das pessoas com deficiência têm **deficiência motora**, mais exactamente $1,5\%$ o que corresponde a 155.476 pessoas. Estes números evidenciam a necessidade de criar sistemas de HSH para apoiar estas pessoas, nas mais diversas **tarefas quotidianas e de integração social e laboral**.

1.2 Enquadramento

Com o objectivo de facilitar o dia a dia destas pessoas idosas, ou com graves limitações funcionais, mais concretamente tetra e paraplégicas, a Universidade de Aveiro, o Centro de Medicina de Reabilitação da Região Centro - Rovisco Pais (CMRRC - Rovisco Pais) e a Micro I/O, criaram uma parceria para desenvolver um **sistema domótico** para apoio à **reabilitação** de pessoas com limitações funcionais graves, o B-LIVE [8]. O sistema domótico **B-LIVE** permite que uma pessoa com limitações funcionais graves possa interagir, de uma forma fácil e cómoda, com diversos **dispo-**

sitivos de uso doméstico, presentes na casa onde habita. Existem diferentes **interfaces** possíveis entre estas pessoas e o B-LIVE. Contudo, em alguns casos, o grau de limitação destas pessoas é tal que torna impossível a sua interacção com o sistema.

O B-LIVE, venceu a edição de 2007 do **Prémio Eng. Jaime Filipe**, um galardão atribuído pelo Instituto da Segurança Social para a melhor concepção inovadora e promotora de autonomia. Este prémio é uma homenagem ao Eng. Jaime Filipe, figura de grande dedicação e actuação na defesa do exercício de cidadania e integração social das pessoas em situação de dependência.

1.3 A fala como interface humano-máquina

A **fala** é vista como um meio de comunicação **natural, eficiente e flexível** entre pessoas [9]. Permite-nos trocar ideias, expressar opiniões, revelar o nosso pensamento. Por outro lado, devido ao desenvolvimento tecnológico dos últimos anos em sistemas de **reconhecimento de fala**, é agora possível controlar dispositivos electrónicos a partir de um computador através de **comandos de fala** [10] [11].

A fala como *Human-Machine Interface* (HMI) é uma alternativa bastante atraente às interfaces actuais (teclado e rato), em particular para pessoas com limitações funcionais [10]. A utilização da fala como HMI apresenta várias vantagens, **não necessita de aprendizagem, permite mãos livres, operação à distância e sem contacto visual**. No entanto, é bastante improvável que nos próximos anos a fala possa substituir definitivamente estes dispositivos. Estudos ergonómicos mostram que interfaces baseados unicamente em reconhecimento de fala não são eficientes devido aos **erros cometidos pelo reconhecedor** [9]. Assim sendo, as interfaces de fala devem complementar as existentes e permitir que o utilizador possa definir qual a interface que melhor se adequa a cada uma das tarefas que pretende realizar. O uso apropriado de fala nos computadores de uso pessoal irá provavelmente requerer o desenvolvimento de um novo conceito de interacção humano-máquina e não apenas modificar as interfaces existentes.

Para as pessoas com limitações funcionais a fala, é nos casos mais problemáticos, a única forma de interacção com as máquinas ao seu redor. **A fala está a ser utilizada como HMI nas mais diversas aplicações**, por exemplo: para controlar cadeiras de rodas eléctricas [12] [13], para

interacção com computadores pessoais [11] e em HSH [14]. Estes projectos são apenas alguns exemplos de como a tecnologia de **reconhecimento de fala pode introduzir benefícios reais na qualidade de vida e independência das pessoas com limitações funcionais**. Existem ainda outros benefícios em utilizar a fala como HMI. Um deles é a possível **integração no mundo do trabalho** de pessoas com limitações funcionais graves. Pessoas que não têm acesso aos computadores, devido às suas limitações, vêem agora uma oportunidade de poderem **realizar as suas tarefas diárias, ou mesmo profissionais**, recorrendo a esta tecnologia. Contudo, tendo em conta os destinatários deste trabalho, existem dificuldades acrescidas em utilizar a fala como HMI. As pessoas com limitações funcionais, devido a lesões ao nível das vértebras **C1, C2 ou C3**, têm **dificuldades ou mesmo incapacidade em respirar** sem auxilio externo. Esta inibição na capacidade de ventilar pode afectar de forma significativa o seu desempenho ao nível da expressão oral, o que pode inviabilizar o uso desta tecnologia.

1.4 Objectivos

O objectivo deste trabalho, é **dotar o sistema domótico B-LIVE com uma interface simples, de fácil utilização e manipulação para pessoas com limitações funcionais, mais propriamente tetra e paraplégicos**. A **fala** apresenta-se como uma **alternativa bastante interessante** às HMI tradicionais, uma vez que, na maior parte dos casos, as limitações destas pessoas não as impedem de se exprimir oralmente. Tendo em conta estes factores, e também com base na experiência dos profissionais de saúde envolvidos, pretendemos dotar o B-LIVE com uma interface baseada em **reconhecimento de fala**.

1.5 Organização da presente dissertação

Esta dissertação encontra-se organizada por capítulos de acordo com a seguinte descrição:

Nesta **introdução** são apresentados os **motivos** que levaram à realização deste trabalho, o contexto em que se insere, assim como os **objectivos a atingir** aquando da sua conclusão. O tema da fala como interface humano-máquina é abordado de uma forma introdutória e muito superficial,

apenas com o objectivo de sensibilizar o leitor para as potencialidades desta nova realidade. No final é apresentada uma breve **descrição da organização da presente dissertação**.

No **segundo capítulo, (”A fala como meio de comunicação entre Humanos”)**, é feita uma breve introdução aos **aparelhos produtor e auditivo dos humanos**. Em relação ao aparelho produtor, pretende-se: identificar os **orgãos envolvidos na produção dos sinais acústicos de fala** e perceber como é que estes sinais são formados; identificar a origem das suas especificidades e classificá-los em função destas. Quanto ao aparelho auditivo, pretende-se: identificar quais os **orgãos envolvidos na recepção do sinal de fala**; perceber o processo de **recepção dos sinais acústicos e posterior percepção pelo cérebro**. Por último, tendo em conta que o público alvo deste trabalho são pessoas com limitações funcionais, é necessário perceber se estas lesões afectam a capacidade destas pessoas se exprimirem oralmente e, em caso afirmativo, de que forma.

No **terceiro capítulo, (”Reconhecimento de fala”)**, faz-se uma breve **introdução aos fundamentos teóricos** que estão na base dos sistemas de reconhecimento de fala. Abordamos os conceitos de **dependência e independência** dos reconhecedores em relação ao orador. Os factores que influenciam o desenvolvimento de projectos com reconhecimento de fala são analisados com algum cuidado. Por fim, apresenta-se a **arquitectura típica** de um reconhecedor de fala, explicando o funcionamento de cada um dos blocos que o constituem. Uma vez que os sistemas de reconhecimento em estudo são baseados em **modelos de Markov não observáveis**, será feita uma breve introdução ao tema.

No **quarto capítulo, (”Desenvolvimento de uma interface *Speech Enabled* para pessoas com limitações funcionais”)**, apresentamos a **interface desenvolvida**. Descrevemos em pormenor o seu princípio de **funcionamento, arquitectura, a base de dados e o reconhecedor de fala que utiliza**. Para percebermos a forma como a interface desenvolvida comunica com o sistema domótico B-LIVE, faz-se uma breve introdução ao B-LIVE na qual se apresentam as suas principais características e também a forma como comunica com o exterior. Ainda neste capítulo, expomos as razões que levaram à escolha da ferramenta *Hidden Markov Model Toolkit* (HTK) para construção do **reconhecedor dependente do orador**. O reconhecedor **independente do orador** será construído com as ferramentas disponibilizadas pelo ***Microsoft Language Development Center* (MLDC)** [15]. Por fim, apresentamos os passos que foi necessário percorrer para construir os reconhecedores utilizados pela interface.

No **quinto capítulo, (”Resultados”)**, serão apresentados os **resultados da avaliação** feita à interface desenvolvida, quer em ambiente **laboratorial, quer em utilização real no CMRRC - Rovisco Pais**.

Finalmente, no **sexto capítulo, (”Conclusões”)**, faz-se um **resumo do trabalho** desenvolvido, abordando as **principais tarefas realizadas**, apresenta-se uma **avaliação dos resultados** obtidos e algumas sugestões para **futuros desenvolvimentos** do trabalho efectuado.

1.6 Resultados já publicados

Depois de concluída, a interface foi testada em ambiente laboratorial. Os resultados obtidos foram submetidos à conferência internacional **DSAI 2007** (*Software Development for Enhancing Accessibility and Fighting Info-exclusion*), sob a forma de **artigo** (com o nome *Speech Enabled Interface to Home Automation for Disabled or Elderly People*) e aceites para publicação nos *proceedings* da conferência. A DSAI 2007 foi organizada pela **Universidade Trás-os-Montes e Alto Douro (UTAD)**, e realizou-se nos dias 8-9 de Novembro de 2007.

Capítulo 2

A fala como meio de comunicação entre Humanos

A comunicação é essencial no dia-a-dia dos seres Humanos, em especial a **comunicação através da fala**. Mais do que qualquer outra característica, é a fala que nos distingue dos animais. Permite-nos **trocar ideias**, **expressar opiniões**, **revelar o nosso pensamento** [16] [17].

Para que possa haver comunicação têm que existir pelo menos três entidades, o **emissor, o receptor e a mensagem**. A comunicação tem início quando o emissor desenvolve uma mensagem (ideia, conceito ou pensamento) que pretende transmitir e termina quando o receptor a descodifica; é no cérebro que a comunicação tem início e termina. À sequência de eventos que ocorrem numa comunicação entre duas pessoas, utilizando a fala, dá-se o nome de **cadeia da fala**, figura 2.1 [16] [18].

A comunicação tem início quando o emissor desenvolve uma mensagem que pretende transmitir. De seguida, a mensagem é codificada e enviada através do sistema nervoso para os músculos que controlam os orgãos do **aparelho produtor**, os quais produzem os sons que são transmitidos sob a forma de **ondas sonoras** para o receptor da mensagem. Do lado do receptor, as ondas sonoras são captadas pelo **aparelho auditivo** e transformadas em **sinais electroquímicos** que são transmitidos para o cérebro. No cérebro, estes sinais são descodificados e analisados no sentido de reconhecer a mensagem enviada pelo emissor [16] [18].

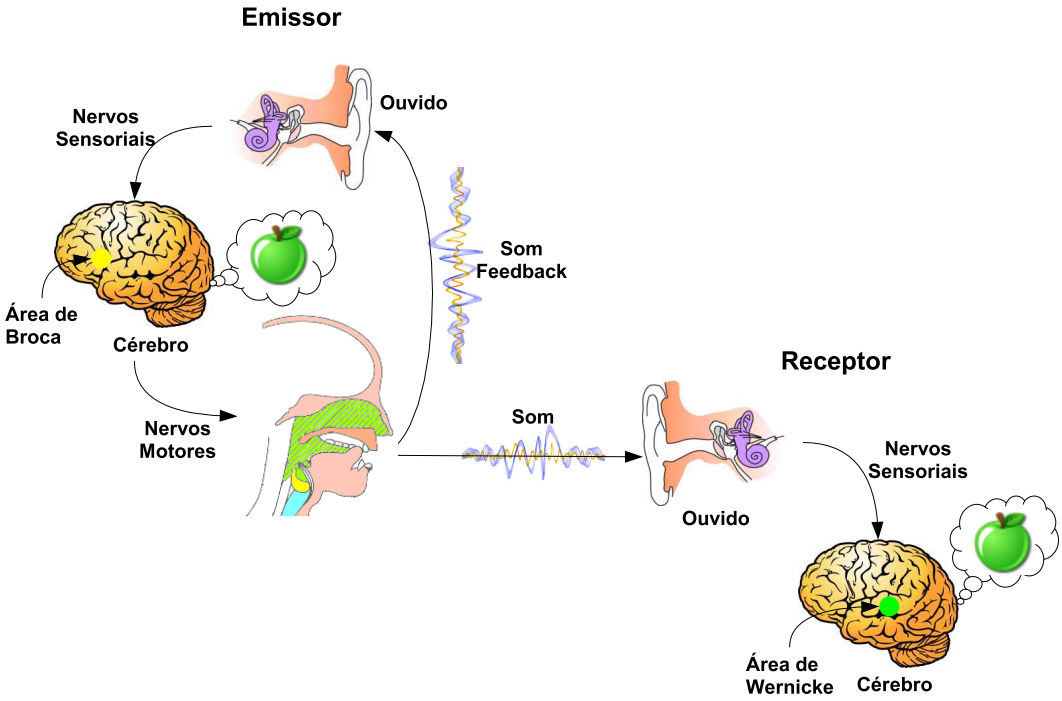


Figura 2.1: Cadeia da fala: produção, transmissão e recepção de sinais de fala (adaptado de várias fontes).

Ao estudarmos a cadeia da fala pretendemos compreender os **mecanismos de produção e recepção dos sinais acústicos** de fala. O objectivo final é perceber de que forma é que as lesões que as pessoas tetra ou paraplégicas apresentam afectam a sua capacidade de produzir e\ou receber estes sinais. Com base no conhecimento obtido sobre o funcionamento dos **aparelhos produtor e auditivo** e a forma como estes são afectados pela tetra ou paraplegia, será possível compreender melhor os possíveis desvios no desempenho dos reconhecedores de fala quando utilizados por pessoas com estas limitações funcionais.

2.1 Produção de sinais acústicos de fala

A disciplina responsável por estudar a produção dos sons da fala é a **Fonética**. O processo de produção da fala, figura 2.2, tem início quando o emissor formula uma mensagem que pretende transmitir. É no cérebro que tudo tem início, mais propriamente na **área de Broca** [19]. O próximo passo é codificar a mensagem de acordo com as regras linguísticas da língua utilizada na comunicação. Esta codificação corresponde em transformar o pensamento numa sequência de

sinais electroquímicos que são transmitidos pelo **sistema nervoso** ao aparelho produtor. No aparelho produtor, os sinais nervosos são utilizados para estimular os diferentes **orgãos e músculos**, como a língua, e os quais ao movimentarem-se, produzem diferentes configurações dos orgãos e, em consequência, diferenças nas ondas sonoras produzidas, de forma a produzir os sons que traduzem a mensagem inicialmente formulada. Por fim, os sons são transmitidos sob a forma de **ondas sonoras** para o receptor da mensagem [16] [17] [18] [19].

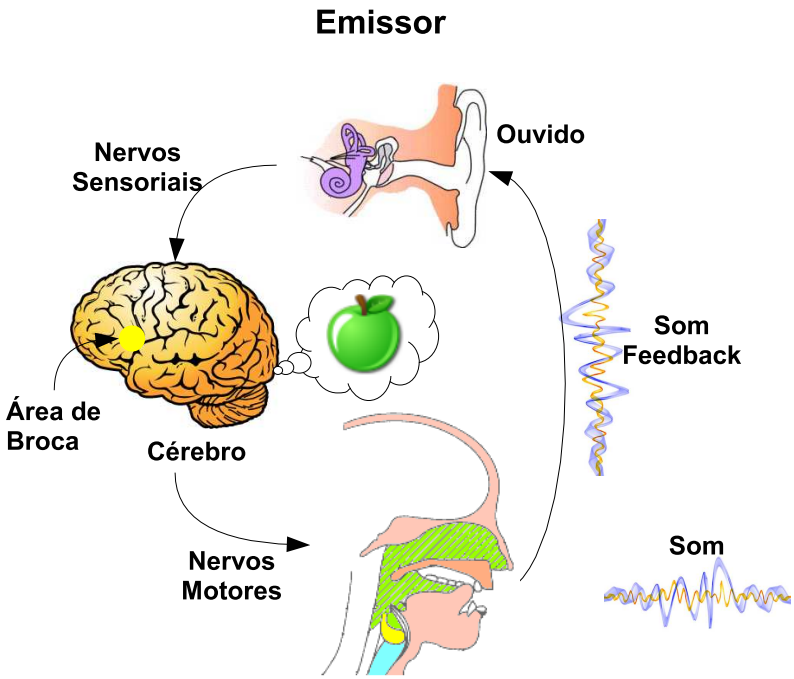


Figura 2.2: Produção da fala. Depois de formular uma mensagem o cérebro envia os sinais nervosos adequados para que os orgãos da fala produzam os respectivos sons (adaptado de várias fontes).

Os orgãos responsáveis pela produção dos sons da fala denominam-se, no seu conjunto, **aparelho produtor**. O aparelho produtor é o responsável pela produção da sequência de **segmentos fonéticos** que constituem os sons da fala [20].

2.1.1 Constituição do aparelho produtor humano

A figura 2.3 representa de uma forma simplificada o aparelho produtor humano [1]. Os seus componentes são: diafragma, pulmões, laringe, faringe, cavidade bucal, língua, dentes, véu palatino ou palato mole, lábios, narinas e cavidade nasal. Normalmente ao conjunto: véu palatino, cavidade oral, língua, lábios e dentes dá-se o nome de **tracto vocal**. À cavidade nasal e parte superior da faringe dá-se o nome de **tracto nasal**. Na análise que se segue serão mencionados apenas os orgãos do aparelho produtor presentes na figura 2.3.

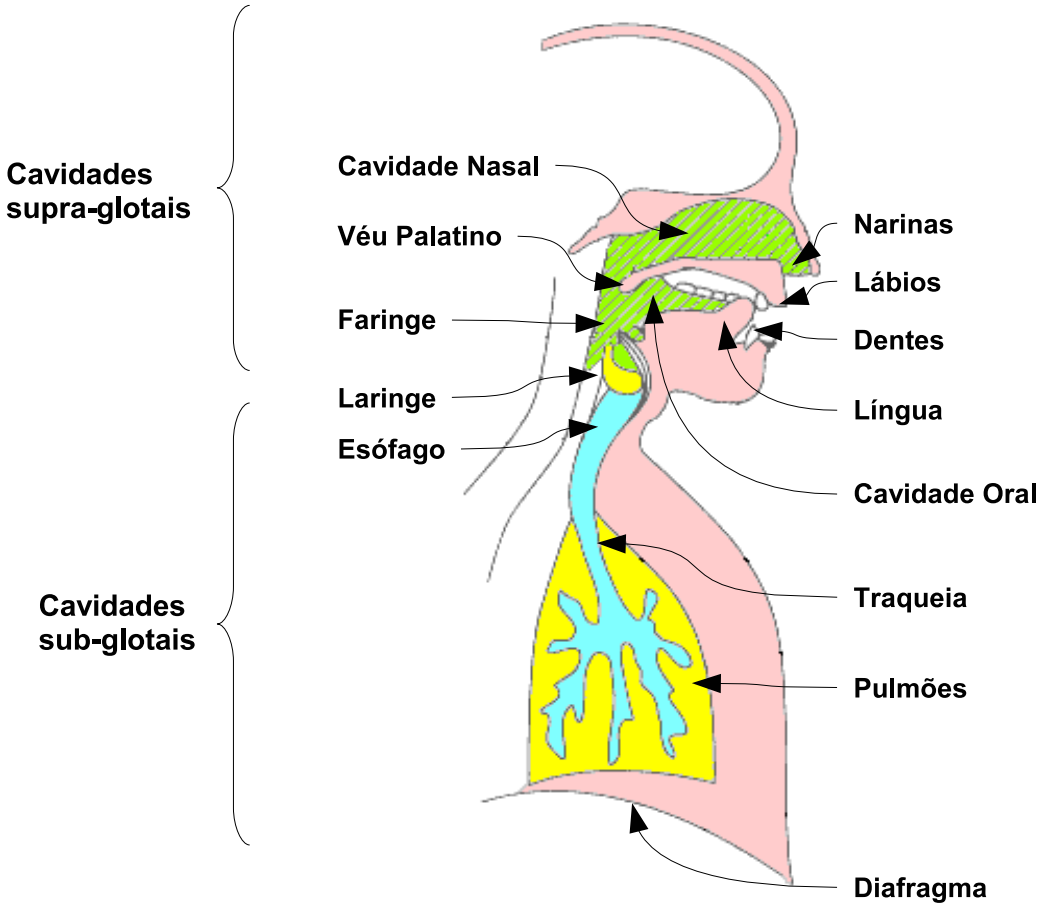


Figura 2.3: Aparelho produtor humano (adaptado de MIT OCW).

Os orgãos que constituem o aparelho produtor podem ser classificados de acordo com a função que desempenham no processo de produção dos sons da fala. Esta classificação permite-

[1] Apenas foram representados na figura os orgãos considerados relevantes para a análise em questão a qual pretendemos que seja simples e objectiva.

nos dividi-lo em **partes**. A tabela 2.1 apresenta de uma forma sucinta estas partes, os órgãos que as constituem e as suas funções [19] [21].

Partes	Componentes	Função
Produtores	Pulmões, músculos abdominais, diafragma, músculos intercostais, músculos extensores da coluna.	Produzem a coluna de ar que pressiona a laringe, produzindo som nas cordas vocais.
Vibrador	Laringe	Produz o som fundamental.
Ressoadores	Cavidade nasal, faringe e cavidade oral.	Ampliam o som.
Articuladores	Lábios, dentes, alvéolos dentários superiores, maxilar inferior, língua, palato duro, palato mole e véu palatino.	Articulam e dão sentido ao som transformando-os em nasais e orais.
Sensor/coordenador	Ouvido – capta, localiza e conduz o som. Cérebro – analisa, regista e arquiva informação relativa ao som.	Captam, seleccionam e interpretam o som.

Tabela 2.1: Partes do aparelho produtor Humano: Produtores, Vibrador, Ressoadores, Articuladores e Sensor/Coordenador.

Os **pulmões**, juntamente com os músculos respiratórios (**diafragma**, músculos intercostais e escalenos e pequeno peitoral), **proporcionam a fonte de energia necessária à produção de fala**, isto é, a corrente de ar ascendente produzida durante a expiração [19] [21]. A intensidade e duração dos sons produzidos dependem da intensidade e volume desta corrente de ar. Quanto maior for a intensidade da corrente de ar ascendente, maior é a intensidade do sinal de fala produzido. Por sua vez, o volume de ar limita a produção de som a um determinado período de tempo. Quanto menor for o volume de ar, menor é o tempo de produção de sons.

A **laringe** é composta por três anéis de cartilagem dentro dos quais estão situadas duas pregas musculares, conhecidas pelo nome de **cordas vocais**. As cordas vocais são pequenos ligamentos com grande poder de contracção e extensão. **São as cordas vocais que, ao vibrarem, produzem os sons da fala**. Se as cordas vocais estiverem juntas (fechadas), a pressão de ar vindo dos pulmões faz com que estas vibrem e dá-se a produção de som ou seja a **fonação** [19].

Os sons produzidos pelas cordas vocais são modulados nas **cavidades oral e nasal**. A cavidade oral, além de amplificar os sons vindos da faringe, possui estruturas anatómicas utilizadas na produção dos sons designadas por **articuladores** (lábios, dentes, alvéolos dentários superiores, maxilar inferior, língua, palato duro, palato mole e véu palatino). Os articuladores podem-se classi-

ficar em **activos** e **passivos**. **Os activos são aqueles que, regra geral, apresentam mobilidade**: os lábios, a língua, o palato mole, o véu palatino e o maxilar inferior. **Os passivos não apresentam mobilidade**. são os dentes, os alvéolos dentários superiores e o palato duro. A classificação dos sons, do ponto de vista articulatório, é feita tendo em conta os articuladores envolvidos na sua produção [19].

2.1.2 Princípios de funcionamento do aparelho produtor humano

De uma forma simples o mecanismo de produção dos sons da fala pode-se descrever do seguinte modo:

O ar vindo dos pulmões (a produção de sons durante a fase de inalação é extremamente rara), através dos **brônquios**, percorre a **traqueia** e o **esófago** até chegar à laringe. Uma vez na laringe, a corrente de ar encontra o seu primeiro obstáculo — a **glote**, que é uma abertura entre as cordas vocais. O fluxo de ar pode encontrá-la fechada ou aberta, dependendo da posição das cordas vocais. Tendo em conta o estado da glote (aberta ou fechada), os sons gerados pelo aparelho produtor podem ser classificados em duas classes distintas: **vozeados ou não vozeados**. Os que são produzidos sem vibração das cordas vocais, com a glote aberta, são designados de não vozeados. Quando a glote se encontra fechada, existe vibração das cordas vocais e os sons assim produzidos são vozeados [19] [20]. A produção de sons resulta da actividade vibratória das cordas vocais. Ao vibrarem, as cordas vocais, aproximam-se e afastam-se alternadamente, gerando uma sucessão rápida de pequenos sopros de ar que, ao passarem por elas produzem os diversos sons, ou seja, dá-se a fonação. Para que as cordas vocais vibrem é necessário que estas estejam suficientemente juntas (fechadas) e que exista uma diferença significativa entre as pressões **subglotal** e **supraglotal**. A pressão subglotal deve ser suficientemente mais elevada do que a pressão supraglotal, para que se estabeleça uma força capaz de vencer a resistência das cordas vocais, fazendo-as afastarem-se uma da outra. Com a abertura das cordas vocais, o ar escapa-se através da glote, o que causa uma diminuição temporária da pressão através deste orgão (**efeito de *Bernouilli***), o que faz com que as cordas vocais se voltem a aproximar. A glote volta à posição inicial (fechada) e o processo repete-se [22].

Ao sair da laringe, a corrente de ar vinda dos pulmões entra na **cavidade faríngea**, uma

encruzilhada que lhe oferece duas vias de acesso ao exterior através dos canais oral e nasal. Entre estes dois canais está o **véu palatino**, orgão dotado de mobilidade capaz de obstruir ou não a passagem do ar pela cavidade nasal e, consequentemente, de determinar a natureza oral ou nasal de um som. Quando levantado, o véu palatino adere à parede posterior da faringe, deixando livre apenas o canal oral. Os sons assim obtidos denominam-se **orais**. Quando baixado, o véu palatino deixa ambos os canais livres e a corrente de ar divide-se, escoando-se uma parte pelas **fossas nasais**. Os sons assim produzidos adquirem o nome de **nasais**. Por fim, é a posição dos orgãos articuladores presentes na cavidade oral que determina o tipo de sons que são gerados [19] [20].

2.1.3 Classificação dos sons da fala

Para que se possa estudar de uma forma sistemática os sons da fala, é necessário representá-los. A escrita não consegue representar de uma forma biunívoca os sons da fala, pelo que é necessário utilizar um conjunto de símbolos adequado para o efeito. Os símbolos utilizados para representar graficamente os sons da fala estão definidos nos **alfabetos fonéticos**. Os mais utilizados são o IPA e o SAMPA, este último está adaptado ao uso em computador [22]. Pelo que, foi o escolhido para a realização deste trabalho. No anexo A apresentamos mais informação acerca destes alfabetos.

A classificação dos sons da fala consiste na sua categorização tendo em conta a observação dos articuladores. Os sons linguísticos classificam-se em **consoantes**, **vogais** e **semi-vogais** [19] [20].

Consoantes

As **consoantes** são produzidas com **constrição** ou **obstrução** significativa à passagem do fluxo de ar no tracto vocal. Tradicionalmente, estas são classificadas segundo dois parâmetros: o modo de passagem do ar pelo tracto vocal — o **modo de articulação** e a região do tracto vocal onde se situa a maior constrição imposta pelos articuladores presentes na cavidade oral — o **ponto de articulação** [19] [20] [22].

O modo de articulação descreve a configuração do tracto vocal devido á posição relativa dos articuladores. Tendo em conta este parâmetro as consoantes podem ser: **oclusivas**, **fricativas**,

laterais, **vibrantes** e **africadas**. As **oclusivas** são produzidas com uma obstrução total à passagem do fluxo de ar pela cavidade oral. As **fricativas** são produzidas com uma obstrução parcial à passagem do fluxo de ar, o que origina turbulência e ruído. Nas **laterais**, a obstrução ao fluxo de ar é provocada pela língua em contacto com o palato ou com os alvéolos, o ar passa pelos lados da língua. Durante a produção das **vibrantes**, existe vibração do órgão articulador - a língua. As **africadas** são produzidas com uma pronuncia mista. No início a obstrução à passagem do fluxo de ar é completa e no final é idêntica à das fricativas. Aplicando estas regras ao Português Europeu, obtemos a classificação apresentada na tabela 2.2 [20] [22] [19].

Modo de Articulação	Vozeadas	Não Voseadas	Nasais
Oclusivas	[b], [d], [g]	[p], [t], [k]	[m], [n]
Fricativas	[v], [z], [Z]	[f], [s], [S]	
Laterais	[l], [L]		
Vibrantes	[r], [R]		
Africadas	[tS], [dZ]		

Tabela 2.2: Consoantes da língua portuguesa e sua classificação, quanto ao modo de articulação. Representadas segundo a nomenclatura SAMPA [23].

O ponto de articulação refere-se à localização do ponto de maior constrição à passagem do ar, imposta pelos articuladores presentes na cavidade oral. Quanto ao ponto de articulação, as consoantes podem ser: **bilabiais** (onde os articuladores são os lábios), **labiodentais** (cujos articuladores são o lábio inferior e os incisivos), **dentais** (os articuladores são a ponta da língua e os incisivos), **alveolares** (onde os articuladores são a ponta da língua e os incisivos superiores), **ápico-alveolares** (cujos articuladores são a ponta ou ápice da língua e os alvéolos), **pré-palatais** (os articuladores são a lâmina da língua e o pré-palato), **palatais** (onde os articuladores são a lâmina da língua e o palato) e **velares** (cujos articuladores são a parte de trás da língua e o véu palatino) [19] [20] [22].

As consoantes podem ainda ser classificadas de acordo com a posição do véu palatino e das cordas vocais. Se o véu palatino estiver afastado da parede da faringe as consoantes são **nasais**, caso contrário são **orais**. Quanto às cordas vocais, estas podem estar abertas (afastadas) ou fechadas (aproximadas), produzindo sons **não vozeados** ou **vozeados**, respectivamente [19] [20] [22].

Vogais e semivogais

As **vogais** e as **semi-vogais** são sons produzidos sem constrição à passagem do ar e com vibração das cordas vocais, pelo que são consideradas sons vozeados [19].

As vogais e as semi-vogais são produzidas sem constrições no tracto vocal, pelo que o fluxo de ar não encontra obstáculos à sua passagem. Desta forma, a classificação das vogais não pode ser feita a partir de pontos de articulação, uma vez que estes não existem. A classificação das vogais é feita segundo os seguintes parâmetros: **posição da língua** (segundo o eixo antero-posterior), **grau de abertura** e **posição dos lábios** [19] [22].

No que diz respeito à posição dos lábios as vogais podem ser **arredondadas** ou **não arredondadas**. As arredondadas são produzidas com **arredondamento dos lábios**, o qual não existe nas não arredondadas [19] [20] [22].

O grau de abertura depende da altura do dorso da língua e da abertura do maxilar inferior no momento de realização da vogal. Tendo em conta o grau de abertura as vogais podem-se classificar da seguinte forma: **abertas**, **semi-abertas**, **semi-fechadas** e **fechadas** [19] [20] [22].

Embora as vogais sejam produzidas com os articuladores abertos, as suas posições são importantes na classificação das mesmas tendo em conta a **região de articulação**. Quanto à posição da língua, esta pode mover-se no sentido antero-posterior (avanço-recuo). As vogais podem ser classificadas como: **anteriores ou palatais**, **médias ou centrais** e **posteriores ou velares**. Em relação à altura da língua as vogais podem ser **altas**, **médias** ou **baixas** [19] [20].

Tendo em conta o papel das cavidades oral e nasal, as vogais podem ser **orais** ou **nasais**, respectivamente [20].

Aplicando este processo de classificação às vogais do Português Europeu, obtemos a classificação apresentada nas tabelas 2.3 e 2.4.

Entre as vogais e as consoantes situam-se as **semi-vogais** ou **glides**, tendo estas as características articulatórias das vogais, mas de duração muito menor. **As semi-vogais nunca ocorrem sozinhas**. Aparecem sempre junto de uma vogal e juntas constituem uma sílaba. Na língua portuguesa apenas existem duas semi-vogais, o [j] e o [w] [19] [20] [22].

Grau de Abertura	Região de articulação			Altura da Língua
	Anteriores	Médias	Posteriores	
Fechadas	[i]	[@]	[u]	Alta
Semi-fechadas	[e]	[6]	[o]	Média
Semi-abertas	[E]		[O]	Baixa
Abertas		[a]		Baixa

Tabela 2.3: Vogais orais da língua portuguesa e sua classificação [20]. Representadas segundo a nomenclatura SAMPA [23].

	Anterior	Média	Posterior
Alta	[i~]		[u~]
Média	[e~]	[6~]	[o~]
Baixa			

Tabela 2.4: Vogais nasais da língua portuguesa e sua classificação [20]. Representadas segundo a nomenclatura SAMPA [23].

2.2 Percepção de fala

O **aparelho auditivo** é responsável pela transformação do som em **impulsos nervosos** que o cérebro descodifica, tornando possível a compreensão da mensagem ouvida. A área do cérebro responsável pela recepção de sinais de fala e compreensão da linguagem é conhecida como **área de Wernicke** [19]. A capacidade de identificar e interpretar a sequência de sons da fala que chega ao ouvido designa-se por **percepção de fala** [19] [21].

O aparelho auditivo desempenha um papel fundamental tanto na fase de produção como na de percepção dos sons da fala. De uma forma bastante simplificada a figura 2.1 ilustra a cadeia de produção e percepção da fala quando existe uma conversação entre dois humanos. A partir desta representação pode-se verificar que os sons da fala captados pelo aparelho auditivo podem ser utilizados para desempenhar funções diferentes, dependendo de quem os recebe [16].

Antes de abordar o funcionamento do aparelho auditivo convém fazer uma pequena abordagem à sua constituição. O ouvido humano pode ser separado em três grandes partes, de acordo com a função desempenhada e a sua localização. São elas: o **ouvido externo** (E), o **ouvido médio** (M) e o **ouvido interno** (I), como ilustra a figura 2.5 [18] [21].

Segue-se então uma pequena descrição do aparelho auditivo humano na qual as suas três

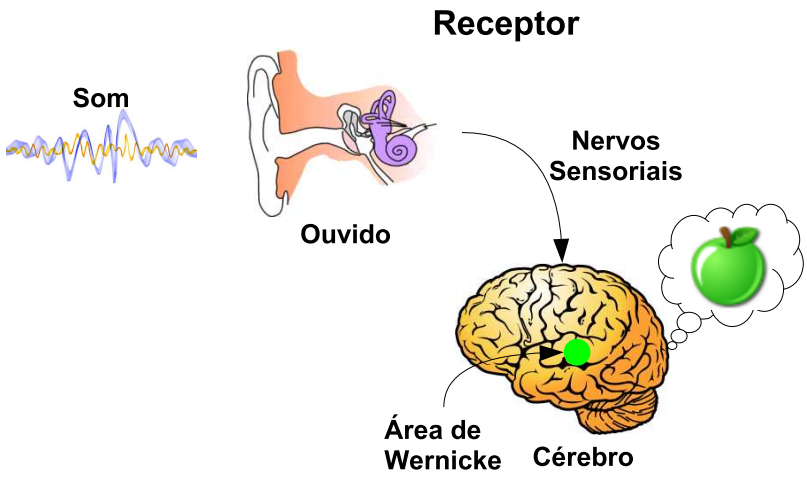


Figura 2.4: Percepção da fala. Os sons são captados e transformados em impulsos nervosos pelo ouvido. De seguida, os impulsos nervosos são analisados e interpretados pelo cérebro com o objectivo de descodificar a mensagem recebida (adaptado de várias fontes).

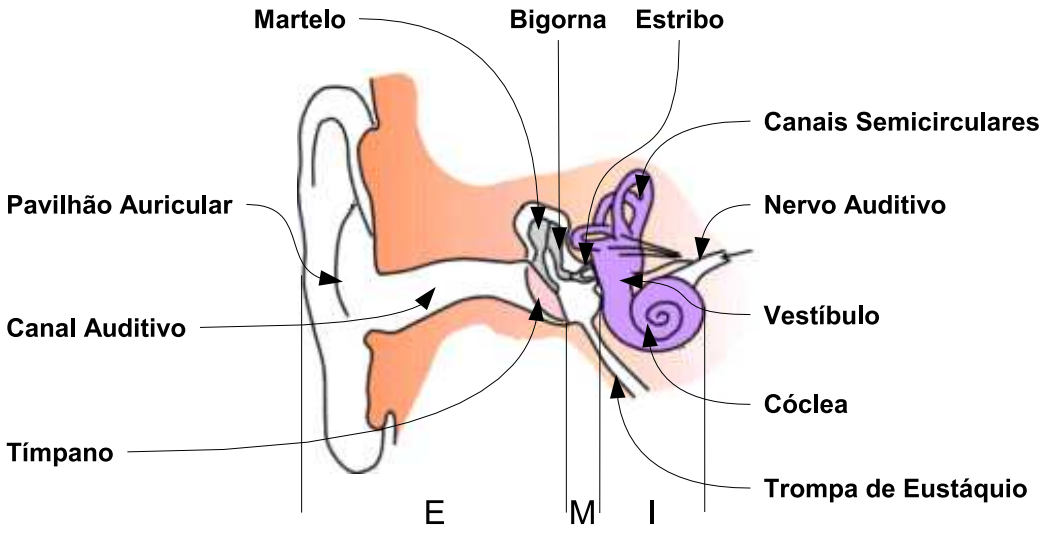


Figura 2.5: Constituição do ouvido humano: ouvido externo (pavilhão auricular, canal auditivo e tímpano); ouvido médio (bigorna, estribo, martelo e a trompa de Eustáquio); ouvido interno (cóclea, vestíbulo, canais semicirculares e nervo auditivo) [21].

zonas constituintes são discriminadas.

2.2.1 Constituição do aparelho auditivo humano

Não é objectivo deste trabalho fazer uma análise exaustiva e muito pormenorizada àcerca do aparelho auditivo e sua constituição. O objectivo é fornecer os elementos necessários à compreensão dos fenómenos associados à recepção e percepção dos sons da fala.

Ouvido Externo

O ouvido externo é constituído pelo **pavilhão auricular** (orelha), pelo **canal auditivo** e pelo **tímpano**. As suas funções são: **captar**, **localizar** e **encaminhar** as ondas sonoras até ao tímpano. O tímpano serve também de **câmara de ressonância**, amplificando algumas frequências. A importância do pavilhão auricular é bem evidente em muitas espécies de mamíferos terrestres. É fundamental na localização de presas e predadores, pelo que é dotado de movimento. Nos humanos esta capacidade foi-se perdendo ao longo da evolução [18] [21].

Ouvido Médio

Fazem parte do **ouvido médio** os **ossículos** e a **trompa de Eustáquio**. É através dele que a energia das ondas sonoras é transmitida do ouvido externo para o **ouvido interno**. A energia é recolhida pelo tímpano e transmitida para o ouvido interno através de três ossos minúsculos, os mais pequenos existentes no corpo humano — o **martelo**, a **bigorna** e o **estribo**. Estes ossículos vibram solidários com o tímpano e transmitem a vibração a uma membrana situada no ouvido interno, a **janela oval**. A trompa de Eustáquio é um canal em parte ósseo, em parte fibrocartilagíneo, existente no ouvido médio. Está em contacto com a **rinofaringe** e tem a função de manter uma pressão constante no ouvido médio [18] [21].

Ouvido Interno

É no **ouvido interno** que se encontra a parte mais importante do aparelho auditivo, sendo constituído pela **cóclea**, pelo **vestíbulo**, pelos **canais semicirculares** e pelo **nervo auditivo**. A **cóclea**, em forma de espiral, é em grande parte responsável pela nossa capacidade de diferenciar e interpretar os sons. **É na cóclea que se desenrola a conversão das ondas sonoras em impulsos eléctricos**. De seguida, estes sinais eléctricos são encaminhados para o cérebro pelo **nervo auditivo**, onde são descodificados e interpretados [18] [21].

2.2.2 Princípio de funcionamento do aparelho auditivo humano

O som pode ser entendido como sendo uma perturbação criada por uma fonte sonora no ambiente que a rodeia. Esta perturbação propaga-se desde a fonte sonora até ao ouvinte, onde é captada pelo seu aparelho auditivo. O pavilhão auricular capta as ondas sonoras e encaminha-as através do canal auditivo para o ouvido médio. **O tímpano vai então vibrar solidário com as moléculas de ar presentes no canal auditivo**. As vibrações captadas pelo tímpano são transmitidas para o interior da cóclea (situada no ouvido interno) através dos ossículos ligados em cadeia entre o tímpano e a janela oval (também situada no ouvido interno). **Os ossículos podem ser vistos como um amplificador**. Actuam como uma alavanca, aumentando a pressão das ondas sonoras. É assim que os sinais sonoros são transmitidos para o interior da cóclea. No seu interior, as vibrações são captadas pelas células ciliadas que identificam as frequências presentes nos sinais sonoros e transmitem a informação correspondente para o cérebro. A transmissão é feita através do nervo auditivo sob a forma de sinais eléctricos. Depois de chegar ao cérebro a informação é descodificada e interpretada, podendo ser utilizada de duas formas diferentes [16] [18] [21]:

Emissor — No emissor o cérebro utiliza os sinais que recebe do aparelho auditivo para controlar o aparelho produtor. Funciona como um mecanismo de ***feedback*** para que o emissor tenha percepção da mensagem que está a produzir [16] [19].

Receptor — Do lado do receptor, os sinais relativos aos sons da fala, captados e transformados em sinais nervosos pelo aparelho auditivo, são descodificados e utilizados pelo cérebro para construir uma ”imagem” alusiva à mensagem que foi recebida [16] [19].

2.3 Coordenação e controlo dos aparelhos produtor e auditivo

Os organismos vivos são sensíveis a alterações ambientais e a estímulos provenientes de diversas fontes internas e externas. O **sistema nervoso** é responsável por **receber**, **transmitir**, **armazenar informações** e **elaborar respostas** adequadas a estes estímulos [18].

A comunicação entre o sistema nervoso e os demais órgãos é feita através de células nervosas chamadas **neurónios**, figura 2.6. Os neurónios são células altamente especializadas, que têm como função transmitir **impulsos nervosos**. As células nervosas estabelecem conexões entre si. Assim, um neurónio pode transmitir a outros os estímulos recebidos, gerando uma reacção em cadeia. Desta forma é assegurada a **integração**, **controlo** e **coordenação** dos diversos sistemas do organismo humano [18].

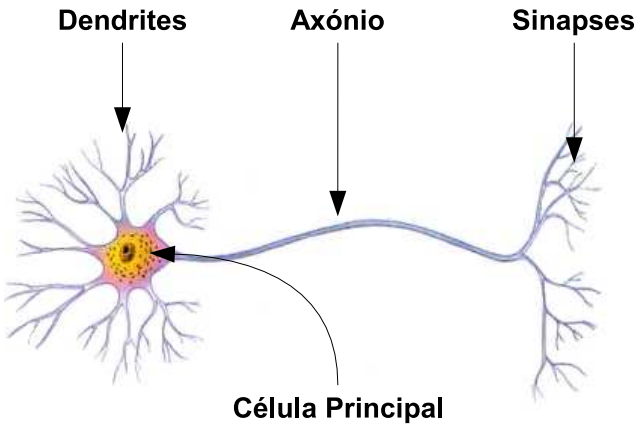


Figura 2.6: Neurónio: célula principal, dendrites, axónio e sinapses

O sistema nervoso tem duas **divisões anatómicas**: o **sistema nervoso central** (SNC) e o **sistema nervoso periférico** (SNP). Fazem parte do SNC o **encéfalo** e a **medula espinal**. Por sua vez o SNP é constituído pelos **nervos** e **gânglios**. Estas divisões anatómicas desempenham diferentes funções. O SNC processa, integra, armazena e responde ao SNP. O SNP é responsável por captar estímulos, transmitir e receber informação para e do SNC [18].

Embora o SNC receba informação sensorial, avalie essa informação e inicie acções sem o contributo do SNP, sozinho ele permaneceria isolado do resto do corpo e do mundo em redor. O SNP recolhe informação de numerosas fontes dentro e fora do corpo e transmite-as ao SNC

através das **fibras aferentes**. As **fibras eferentes** do SNP transmitem a informação do SNC para as várias partes do corpo, primariamente para os músculos e glândulas, regulando a actividade destas estruturas. **Sem o SNP, o SNC não receberia informação e seria incapaz de produzir respostas observáveis**. Nem mesmo os pensamentos e emoções poderiam ser expressos por causa do isolamento do SNC [18].

O SNP pode ser dividido em duas partes: uma parte **craniana**, que consiste em doze pares de nervos, e uma parte **espinal**, constituída por trinta e um pares de nervos. No caso particular deste trabalho interessa estudar a parte craniana pois é ela que vai enervar os aparelhos produtor e auditivo [18].

Por convenção os nervos cranianos são numerados em numeração romana, de I a XII, do mais **anterior** para o mais **posterior**. A figura 2.7 é uma representação dos nervos cranianos, parte **aferente** e **eferente**.

Os nervos cranianos podem ser de três tipos **sensoriais**, **motores** e **mistos**. A tabela 2.5, apresenta a sua classificação e apresenta uma breve descrição das funções de cada um deles [21].

Nervo	Tipo	Função
I. Olfactivo	Sensorial	Olfactiva
II. Óptico	Sensorial	Visão
III. Oculomotor	Motor	Controlo dos músculos do globo ocular, da pupila e do cristalino
IV. Troclear	Motor	Controlo dos músculos do globo ocular
V. Trigémeo	Misto	Sensitiva, para o controlo facial e para a sensação do gosto Motora, para a mastigação
VI. Abducente	Motor	Controlo dos movimentos do globo ocular
VII. Facial	Misto	Sensitiva, para o gosto Motora, para os músculos faciais e glandulossalivares
VIII. Auditivo	Sensorial	Auditiva e equilíbrio
IX. Glossofaríngeo	Misto	Sensitiva, para o gosto Motora, para a mastigação
X. Vago	Misto	Sensitiva, para a faringe, laringe e vísceras torácicas e abdominais Motora, para a faringe e laringe
XI. Acessório	Motor	Controlo da faringe, laringe e do palato
XII. Hipoglosso	Motor	Controlo dos músculos da língua

Tabela 2.5: Classificação dos nervos cranianos e respectivas funções [21].

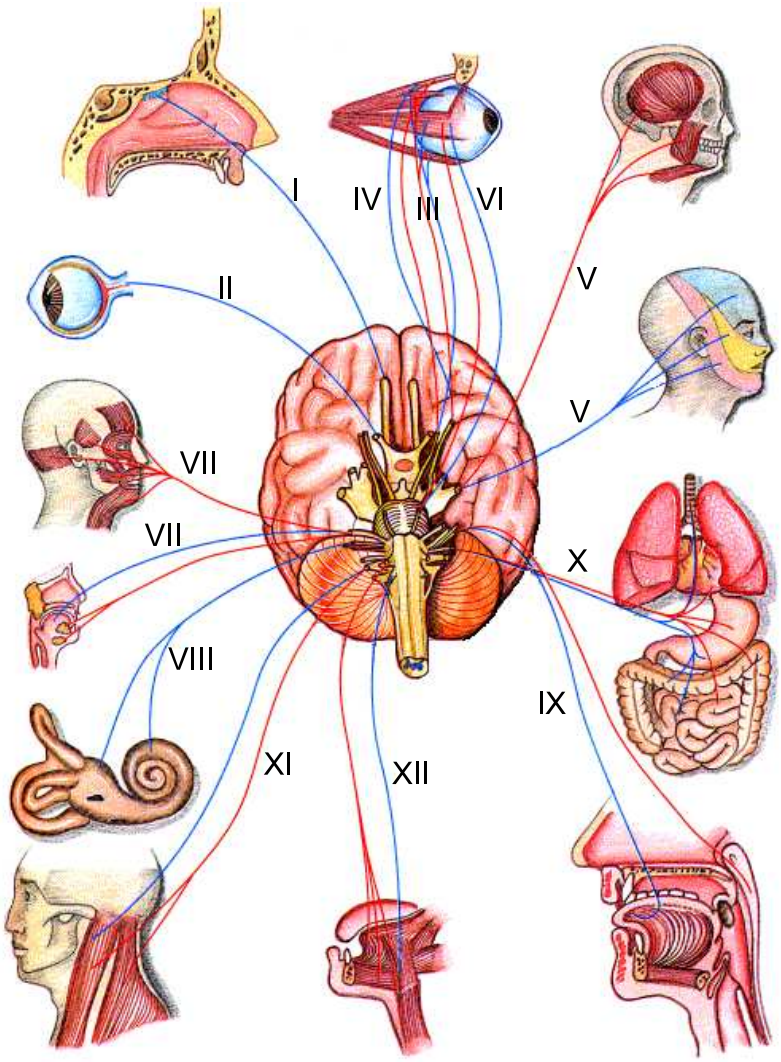


Figura 2.7: Nervos cranianos. I-Nervo Olfactivo, II-Nervo Óptico, III-Nervo Oculomotor, IV-Nervo Troclear, V-Nervo Trigémio, VI-Nervo Abducente, VII-Nervo Facial, VIII-Nervo Auditivo, IX-Nervo Glossofaríngeo, X-Nervo Vago, XI-Nervo Acessório, XII-Nervo Hipoglosso [18] [21] [24].

2.3.1 Enervação do aparelho produtor humano

Não se pode dizer que existe um nervo responsável pela enervação dos vários orgãos e músculos envolvidos no processo de geração dos sons da fala. Praticamente todos os nervos cranianos têm um papel mais ou menos importante neste processo. São importantes os nervos responsáveis pela enervação dos músculos faciais (nervo facial), pelo movimento da língua (nervo hipoglosso) e dos maxilares (nervo trigémio). **Os orgãos mais directamente envolvidos na produção de sons (palato mole, faringe, músculos intrínsecos da laringe) são enervados pelo nervo vago** [21].

Os sinais nervosos transportados pelos nervos cranianos não passam pela medula espinal, pelo que uma lesão a este nível não afecta a capacidade das pessoas tetraplégicas produzirem sons.

2.3.2 Enervação do aparelho auditivo humano

A função sensorial relativa à audição é assegurada pelo nervo auditivo. O nervo auditivo divide-se em duas partes, **uma vestibular e outra coclear**. O termo vestibular refere-se ao vestíbulo do ouvido interno, envolvido no equilíbrio. O termo coclear refere-se à cóclea, a porção do ouvido interno envolvida na audição [21].

A informação sensorial relativa ao sentido da audição é transportada até ao córtex auditivo sem percorrer a espinal medula. **Uma lesão no SNC ao nível da medula espinal não tem, ou tem pouco impacto na audição**.

2.4 Comentários finais

A qualidade de muitos sons da fala pode ser bastante modificada por alterações na configuração e, consequentemente, nas propriedades acústicas do trato vocal. Essas mudanças são provocadas principalmente por alterações na forma da cavidade oral, por exemplo devido à falta de dentes ou à colocação de aparelhos dentários.

As lesões na medula espinal que provocam a tetra e paraplegia não têm influência di-

recta no funcionamento dos aparelhos produtor e auditivo. Isto acontece porque estes aparelhos são enervados pelos nervos cranianos que não são afectados por lesões ao nível da medula espinal. No entanto pode acontecer que a capacidade de produzir sons seja afectada por lesões do aparelho respiratório ou por intervenções médicas, por exemplo uma toracotomia[2].

Assim, uma pessoa com tetra ou paraplegia pode apresentar alguma dificuldade em produzir sons, sendo as mais notórias: **o cansaço, a rouquidão, a baixa amplitude dos sons produzidos e a dificuldade em colocar a voz**.

[2] Abertura cirúrgica da cavidade torácica.

Capítulo 3

Reconhecimento de fala

O processamento de fala tem vindo a ganhar grande importância nos últimos anos, em parte devido aos resultados da investigação que tem vindo a ser realizada na área, mas também devido aos avanços tecnológicos que permitem uma cada vez **maior capacidade de processamento e armazenamento de dados**. O reconhecimento de fala, um dos ramos do processamento, não é excepção e apresenta também um enorme crescimento não só ao nível de conhecimento adquirido, mas também da quantidade e qualidade dos sistemas de reconhecimento disponíveis.

Os sistemas de reconhecimento de fala sofreram um enorme desenvolvimento nas últimas décadas. **A redução da taxa de erro de palavra e diminuição do tempo de processamento** necessário para fazer o reconhecimento, resultaram em sistemas mais fiáveis e possibilitaram que estes saíssem dos laboratórios onde foram desenvolvidos para serem utilizados em aplicações reais.

O nível de desenvolvimento existente não teria sido possível sem a introdução de modelos matemáticos e estatísticos nos sistemas de reconhecimento de fala, nomeadamente, a utilização de *Hidden Markov Models* (HMM's) para modelar o sinal de fala. **Os HMM's são a base teórica que está por trás dos mais avançados sistemas de reconhecimento de fala existentes na actualidade**. Estes permitem modelar as variações temporais e espectrais em simultâneo [25]. Os parâmetros para construção destes modelos podem ser obtidos automaticamente a partir de procedimentos e dados de treino. O processo de treino é fundamental para obter modelos que permitam

realizar reconhecimento com uma taxa de sucesso elevada. A qualidade dos dados disponíveis para treino dos sistemas é também um factor bastante importante, pelo que foi feito um esforço no sentido de desenvolver **grandes bases de dados de fala para investigação, desenvolvimento, treino e avaliação dos sistemas de reconhecimento de fala**.

Um outro factor importante foi o **estabelecimento de normas** para a avaliação do desempenho. Quando os investigadores começaram a desenvolver os seus reconhecedores, utilizavam dados de fala recolhidos nos seus próprios laboratórios, não obedecendo a critérios de selecção bem definidos. Em consequência, não era possível comparar o desempenho dos reconhecedores dos diferentes laboratórios. A recente disponibilidade de grandes bases de dados de domínio público, associada à especificação de rigorosos critérios de avaliação, resultou num rigor e aceitação dos resultados obtidos em diferentes laboratórios.

3.1 Definição do problema

Um sistema de reconhecimento automático de fala é um sistema capaz de, pelo menos, identificar várias palavras ou frases quando proferidas oralmente por um determinado indivíduo na ausência de qualquer outro sinal acústico. Idealmente, seria também capaz de transcrever qualquer discurso oral, pelo menos nas circunstâncias de audição consideráveis aceitáveis por um ouvinte humano. Neste contexto, considera-se como dados para o reconhecimento, apenas o sinal acústico resultante do processo da fala [26].

Na avaliação dos projectos de sistemas de reconhecimento de fala é necessário, antes de mais, determinar o fim a que se destinam. Um sistema que pretende transformar comandos vocais em instruções a que uma máquina deve obedecer, é menos exigente do que um sistema que pretende transformar em texto sequências reais de fala. Por exemplo, um sistema de comandos vocais é bastante limitado em termos de vocabulário, exigindo apenas, em média, algumas dezenas de palavras, correspondentes aos comandos a executar. Pelo contrário, num sistema em que o objectivo é o reconhecimento de sequências reais de fala, são exigidas em média, dezenas ou centenas de milhar de palavras [27].

Um outro factor que condiciona desde o início o projecto de sistemas de reconhecimento de

fala, é a forma como vão ser utilizados. Isto é, o reconhecedor vai ser utilizado apenas por uma única pessoa, ou por várias? Se um sistema de reconhecimento de fala se destina ao uso exclusivo de um único orador, poderá ser **dependente do orador**. Se pelo contrário, se destinar ao uso de um grupo mais ou menos vasto de oradores, em que não é possível identificar cada um de modo a atribuir-lhe um reconhecedor específico, então este deverá ser **independente do orador**. Numa situação intermédia consideram-se reconhecedores *multi-orador*, destinados a um grupo especifico de oradores. Em geral, obtêm-se melhores resultados no reconhecimento quando se treina um reconhecedor para ser utilizado apenas por um único orador, contudo o esforço requerido ao orador para o treino do ”seu reconhecedor” é, em muitos casos, excessivo, sobretudo se este não estiver devidamente motivado para o efeito. Além disso, **um reconhecedor dependente do orador apresenta um desempenho medíocre, quando confrontado com qualquer outro orador diferente daquele para o qual foi treinado**. A solução utilizada nos reconhecedores independentes do orador consiste no treino dos modelos com um corpus de fala com um número elevado de oradores, considerados representativos de uma determinada população. Desta forma, obtêm-se resultados de reconhecimento aceitáveis com oradores não utilizados no treino do reconhecedor. Ainda assim, apresentam obviamente um desempenho inferior ao dos reconhecedores concebidos exclusivamente para um grupo ou orador específico.

O desenvolvimento de sistemas para reconhecimento de fala é extremamente dificultado pela variabilidade do respectivo sinal acústico. Esta variabilidade é devida a factores muito diversos, tais como: **entoação, tom de voz, o estilo do discurso e sotaque**, entre outros. Em adição a estes factores devidos a quem produz o sinal de fala, existem outros que variam com o ambiente em redor do orador, **ruído ambiente** inerente ao espaço onde o orador se encontra, mas também, de **conversas paralelas** que possam existir entre outros oradores presentes no mesmo espaço. Perante esta panorâmica pode-se esperar a existência de uma infinidade de sinais de fala, pelo que, é fácil compreender a necessidade de restringir, tanto quanto possível, a influência de alguns destes factores no sinal, por forma a obterem-se modelos de complexidade e dimensões aceitáveis.

No reconhecimento, os aspectos da variabilidade do sinal de fala exclusivamente devidos às características do orador são considerados separadamente em duas classes: **a variabilidade intra-orador e a variabilidade inter-orador**. Nos reconhecedores dependentes do orador, interessa,

essencialmente, atenuar os efeitos da primeira, enquanto que nos reconhecedores independentes do orador interessa atenuar a segunda.

A variabilidade intra-orador refere-se a variações temporais das características de um dado orador. Estas são devidas a alterações de dois tipos [28]:

Físicas — estão relacionadas, essencialmente, com a condição física do orador. Uma simples constipação, ou outra patologia que possa afectar o trato vocal, pode alterar as características do sinal de fala.

Emocionais — as alterações do estado emocional do orador podem ser de diversos tipos: alegria, tristeza, admiração, entre outras. Estas ocorrem com mais frequência e mais rapidamente do que as do tipo físico.

A variabilidade inter-orador pode ser relacionada com as inúmeras formas de classificar ou diferenciar os seres humanos, em termos físicos, psicológicos, comportamentais, sociais, económicos, religiosos, geográficos, etc. Todas estas diferenças impõem características especificas ao processo de produção de fala que são identificáveis no respectivo sinal. Os factores de variabilidade inter-operador mais relevantes para o reconhecimento de fala são: **a idade, o sexo, o peso, o hábito de fumar, o nível cultural, o sotaque, etc.**. Pode-se então concluir que as diferenças no sinal de fala produzido por diferentes oradores estão relacionadas não só com a configuração do seu trato vocal, mas também com hábitos linguísticos adquiridos por motivos diversos [28].

A variabilidade relativa às condições ambientais está intimamente ligada ao processo de captação do sinal de fala. O ruído devido ao meio em que se encontra o orador é caracterizado em função de determinados ambientes típicos. Nos espaços públicos exteriores coexistem vários ruídos sobrepostos, tais como, o de veículos na via pública e o **burburinho citadino**. Nas salas de grandes dimensões, onde é grande a afluência de pessoas, surgem **efeitos de reverberação**. Num ambiente de escritório ou doméstico, geralmente os espaços são mais pequenos e os ruídos são em geral do tipo impulsivo, gerados por fontes muito próximas: máquinas registadoras, impressoras, teclados, telefones, campainhas, electrodomésticos, ou mesmo fala de outros oradores [28]. Para reduzir o efeito deste tipo de ruído existem várias técnicas, entre as quais se destacam:

Subtracção espectral — esta técnica é particularmente eficaz na atenuação de ruído quase estacionário [26].

Cancelamento adaptativo de ruído — utiliza dois ou mais microfones, um para captar o sinal de fala corrompido por ruído e os restantes para captarem o próprio ruído [26].

Este tipo de soluções são, contudo, pouco eficazes quando o ruído é do tipo impulsivo. Numa perspectiva mais vasta e integrada com o próprio reconhecedor, surgem alternativas como a da generalização dos modelos de Markov não observáveis convencionais, para uma decomposição óptima de processos simultâneos. Com o uso de modelos perceptuais, que resultam da modelação dos fenómenos acústicos, fisiológicos e psicológicos que ocorrem no ouvinte humano, têm-se conseguido melhorias no desempenho dos sistemas de reconhecimento de fala [26].

De tudo o que foi dito, facilmente se verifica o **carácter interdisciplinar presente no projecto de sistemas de reconhecimento de fala**. O reconhecimento de fala tem por base o conhecimento científico das áreas do processamento de sinais, do reconhecimento de padrões e linguística. Contudo, este tema não se esgota nestas duas áreas do conhecimento, abrangendo muitas outras.

3.2 Componentes de um reconhecedor típico

O reconhecimento automático de fala pode ser encarado como um problema de descodificação, ou seja, como encontrar a sequência de palavras que corresponde à sequência de elocuções observada. Os sistemas de reconhecimento existentes assumem que o sinal acústico correspondente ao sinal de fala de entrada, corresponderá à sequência de palavras mais provável, avaliada pelo sistema, segundo os modelos acústicos e de linguagem adoptados.

O **sinal acústico** correspondente à elocução é representado por uma sequência de **vectores de parâmetros** extraídos do sinal de fala [29], conforme ilustra a figura 3.3. A sequência de vectores de parâmetros extraídos do sinal acústico é representada por $O = \{O_1, O_2, ..., O_T\}$.

Para encontrar a **sequência de palavras** $\hat{W} = \hat{w}_1, \hat{w}_2, ..., \hat{w}_N$, correspondente ao sinal acústico

de entrada, aplica-se o critério de **máxima probabilidade à posteriori**:

$$\hat{W} = \arg\max_{W} P(W|O) \tag{3.1}$$

Aplicando a **regra de *Bayes*** podemos reescrever a equação 3.1 da seguinte forma:

$$\hat{W} = \arg\max_{W} \frac{P(O|W)P(W)}{P(O)} \tag{3.2}$$

O termo $P(O)$ é constante para qualquer sequência de palavras avaliada, pelo que pode ser removido. Assim sendo, a sequência de palavras $\hat{W}$ corresponderá à sequência W que maximiza o produto $P(O|W)P(W)$, isto é:

$$\hat{W} = \arg\max_{W} \{P(O|W)P(W)\} \tag{3.3}$$

O termo $P(O|W)$ é avaliado pelo **modelo acústico** e representa a probabilidade dos modelos que representam a sequência de palavras W gerarem a sequência de observações O. Neste trabalho vamos abordar apenas os modelos acústicos com base em modelos de Markov não observáveis. O termo $P(W)$ é avaliado pelo **modelo da linguagem**. Este consiste na probabilidade à priori de observar a sequência de palavras W e é independente da sequência de vectores observados.

Agora que já conhecemos quais os factores que influenciam a busca pela sequência de palavras que representa o sinal acústico observado, estamos em condições de apresentar o **diagrama geral de um reconhecedor de fala**, o qual é representado pela figura 3.1. Neste capítulo vamos apresentar, de uma forma muito sucinta, cada um dos blocos nela representados.

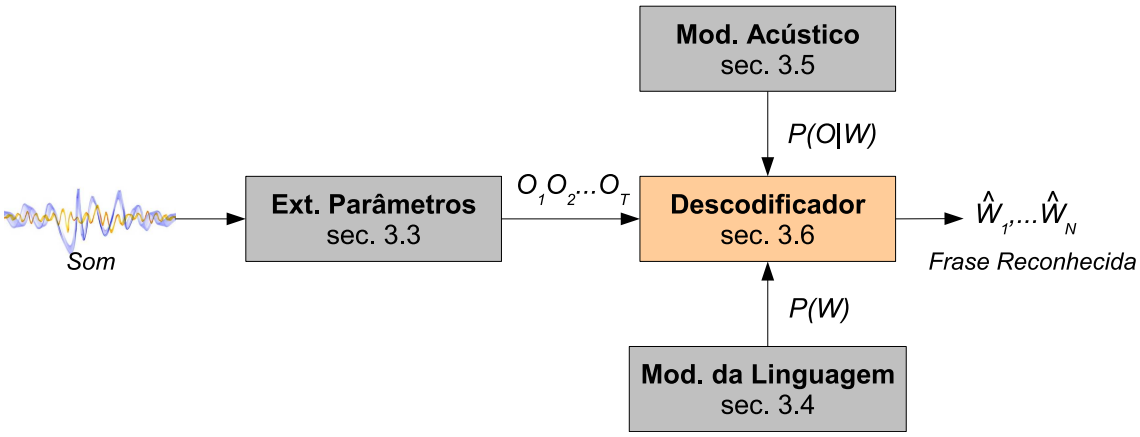


Figura 3.1: Diagrama típico de um sistema de reconhecimento de fala [30]

3.3 Extracção de parâmetros

O objectivo deste ponto é descrever como **transformar o sinal acústico de fala numa sequência de vectores de parâmetros**, cada um deles representativo de um pequeno fragmento do sinal. Em reconhecimento de fala, os parâmetros mais utilizados são os ***Mel Frequency Cepstral Coefficients*** (MFCC). A figura 3.2 apresenta os passos necessários à extracção destes coeficientes [30]. Nesta descrição é assumido que o sinal acústico de fala já se encontra devidamente **amostrado e quantificado**.

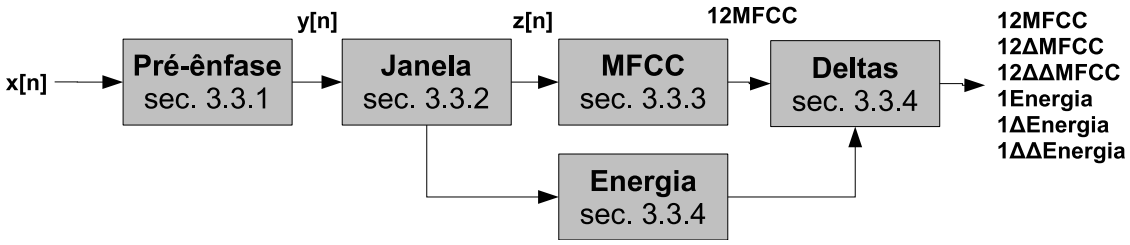


Figura 3.2: Processamento do sinal de áudio.

3.3.1 Pré-ênfase

Devido à natureza do aparelho produtor humano **o sinal acústico da fala tem mais energia às baixas do que às altas frequências**, o que pode levar à criação de modelos acústicos que não têm (ou têm pouco) em conta a informação presente nos formantes de mais altas frequências [30]. Para evitar que isto aconteça, é comum proceder à **suavização do espectro do sinal acústico**. A esta operação dá-se o nome de pré-ênfase. Em reconhecimento de fala a pré-ênfase é feita através de um filtro digital de primeira ordem:

$$y[n] = x[n] - \alpha x[n-1] \tag{3.4}$$

onde $x[n]$ representa a amostra no instante n do sinal de entrada e $0.9 < \alpha < 1.0$ [30].

3.3.2 Aplicação da janela de análise

Como já foi dito anteriormente, **o sinal acústico de fala é representado por um conjunto de vectores de parâmetros**. Cada um destes vectores representa uma pequena porção do sinal, à qual

se dá o nome de ***frame***. **O sinal acústico de fala caracteriza-se como sendo não estacionário, isto significa que as suas características variam ao longo do tempo**. Contudo, se considerarmos pequenas *frames*, tipicamente com duração de aproximadamente 25 ms, **podemos assumir que dentro de cada *frame* ele é estacionário** [29].

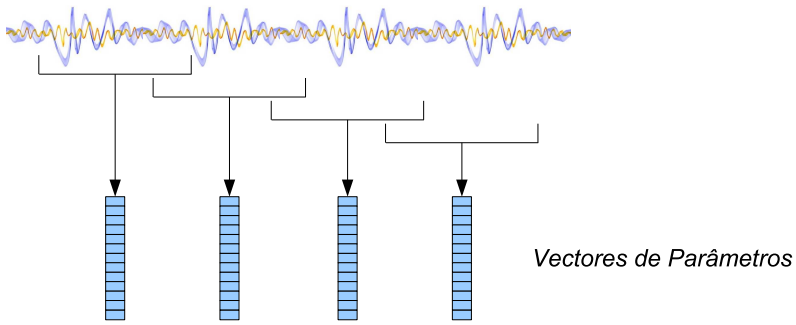


Figura 3.3: Extracção dos vectores de parâmetros do sinal de fala.

A obtenção das *frames* é feita através da deslocação de uma **janela de *Hamming*** ao longo do sinal, a qual é feita com um **incremento de 10 ms**, ver a figura 3.3. A equação 3.5 define uma janela de Hamming típica [29].

$$h[n] = \begin{cases} 0.54 - 0.46\cos(\frac{2\pi n}{N-1}) & \text{para} \quad 0 \leq n \leq N-1 \\ 0 & \text{para} \quad outros \end{cases} \tag{3.5}$$

Onde N é o número de amostras correspondente ao comprimento da janela, neste caso é o número de amostras contidas em cada frame de 25 ms.

O sinal z, à saída deste bloco, obtém-se multiplicando o valor do sinal y no instante n pelo valor da janela no mesmo instante, isto é:

$$z[n] = h[n]y[n] \tag{3.6}$$

3.3.3 Coeficientes *Mel Frequency Cepstral Coefficients*

A figura 3.4 ilustra as etapas que são necessárias percorrer para **extrair os coeficientes MFCC de uma *frame* de sinal acústico de fala**. O primeiro passo é a **análise espectral**. Normalmente, aplica-se o algoritmo da ***Fast Fourier Transform*** (FFT) para obter o espectro do sinal z. O algoritmo da FFT é bastante eficiente do ponto de vista computacional, contudo, apenas se pode aplicar se o comprimento do sinal z for uma potência de dois, isto é $N = 2^p$ com p inteiro [30].

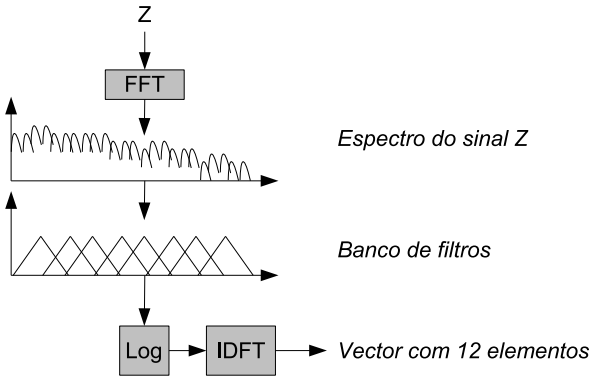


Figura 3.4: Extracção dos coeficientes MFCC.

O espectro obtido a partir da aplicação da FFT, pode ser entendido como sendo informação sobre a quantidade de energia em cada uma das frequências presentes no sinal acústico de fala em análise. **Sabe-se que o ouvido humano não tem a mesma sensibilidade a todas as frequências**. É menos sensível às altas frequências, tipicamente acima dos 1000 Hz. Uma forma de modelar esta característica é submeter o espectro a um **banco de filtros triangulares** separados entre si segundo a **escala *mel***. **A escala *mel* proposta por Stevens, Volkmann e Newmann em 1937, mede a sensação subjectiva de tom (”*pitch*”) em função da frequência**, equação 3.7. Esta escala é linear abaixo dos 1000 Hz e logarítmica acima deles [30].

$$mel(f) = 1127ln(1 + \frac{f}{700}) \tag{3.7}$$

De uma forma geral, **a resposta do ouvido humano à intensidade dos sinais acústicos é logarítmica**. Para modelar este comportamento, aplica-se o logaritmo a cada um dos coeficientes *mel* extraídos pelo banco de filtros. Esta operação tem ainda a vantagem de minimizar variações bruscas do sinal acústico.

Por último, para obter os coeficientes cepstrais, aplica-se a ***Inverse Discrete Fourier Transform*** (IDFT) aos 12 primeiros coeficientes *mel*.

3.3.4 Energia e coeficientes delta

Aos 12 coeficientes mel-cepstrais já existentes junta-se ainda informação sobre a energia presente no sinal. A energia presente no sinal é um dado importante para detecção de sinal acústico re-

levante, pois, **os sinais correspondentes às vogais e silabas são mais energéticos que as pausas**. A energia contida em cada *frame* pode ser calculada da seguinte forma:

$$E = \sum_{n=0}^{N-1} z^2[n] \tag{3.8}$$

A estes 13 coeficientes juntam-se ainda os coeficientes delta de primeira e segunda ordem. Estes **coeficientes revelam como é que os coeficientes mel-cepstrais variam e a que ritmo**. Por outras palavras, são uma medida sobre a velocidade e aceleração dos coeficientes mel-cepstrais. O resultado é um vector com 39 elementos contendo 12 coeficientes mel-cepstrais, a energia presente na frame, 13 coeficientes delta de primeira ordem e 13 coeficientes delta de segunda ordem.

3.4 Modelo da linguagem

O modelo da linguagem [1] pode ser encarado como um **conjunto de restrições à combinação das palavras presentes nas frases reconhecidas**. Estes podem ser **estatísticos ou determinísticos**.

O modelo estatístico da linguagem permite calcular a **probabilidade à priori da ocorrência de uma determinada sequência de palavras** $w_1, w_2, ..., w_K$. Se representarmos a probabilidade conjunta de ocorrência de uma sequência de K palavras por $P(W_1^K)$ esta pode ser calculada da seguinte forma [30]:

$$P(W_1^K) = P(w_1) \prod_{k=2}^{K} P(w_k | w_1 ... w_{k-1}) \tag{3.9}$$

3.4.1 Modelos N-grams

Para simplificar o cálculo da probabilidade $P(W_1^K)$ e também para diminuir a carga computacional, pode-se assumir que a escolha da palavra w_k não depende de toda a sequência passada $w_1, w_2, ..., w_{K-1}$ mas sim das $n-1$ palavras anteriores a ela. Os modelos **N-grams** baseiam-se neste pressuposto. Exemplificando para $n = 2$ e $n = 3$ temos os modelos bigram e trigram repre-

[1] A palavra *Language* pode ser traduzida de duas formas: Linguagem ou Língua. No contexto dos sistemas de reconhecimento de fala é usual utilizar Linguagem.

sentados pelas equações 3.10 e 3.11, respectivamente.

$$P(W_1^K) = P(w_1) \prod_{k=2}^{K} P(w_k|w_{k-1}) \tag{3.10}$$

$$P(W_1^K) = P(w_1)P(w_2|w_1) \prod_{k=3}^{K} P(w_k|w_{k-2}, w_{k-1}) \tag{3.11}$$

3.4.2 Modelos *Finite State Model*

Neste trabalho, o modelo da linguagem utilizado é do tipo ***Finite State Model*** (FSM). **A utilização destes modelos apenas é possível quando o vocabulário é pequeno**. Neste caso, a gramática não é mais do que uma **rede de palavras** que correspondem às frases que se pretendem reconhecer, como mostra a figura 3.5. Neste caso, a procura de sequências de frases válidas resume-se à sua validação numa **máquina de estados**.

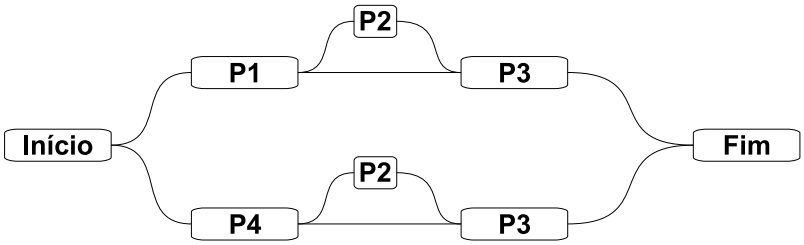


Figura 3.5: Gramática do tipo FSM.

Podem também ser utilizadas **gramáticas livres de contexto** (*Context-Free Grammars*) (CFG's), para gerar modelos FSM. Uma CFG consiste num conjunto de regras que determinam como é que as palavras de uma determinada linguagem se podem agrupar e ordenar [30].

3.4.3 Perplexidade

A comparação entre diferentes modelos da linguagem é bastante importante. Desta forma, pode-se ter uma medida da eficiência que estes impõem aos sistemas de reconhecimento. A forma mais correcta de avaliar os modelos da linguagem é introduzi-los em sistemas de reconhecimento e avaliar a taxa de erro global. Contudo, este método é bastante dispendioso [30].

Domínio	Perplexidade
Radiologia	20
Medicina de emergência	60
Jornalismo	105
Fala geral	247

Tabela 3.1: Perplexidades típicas para diferentes domínios.

A medida mais utilizada na avaliação de modelos da linguagem é a perplexidade. A perplexidade pode ser vista como uma **medida aproximada do factor de ramificação de um modelo**. O factor de ramificação de um modelo é o número de palavras que se podem seguir a qualquer palavra. Formalmente a perplexidade é definida da seguinte forma [30] [28]:

$$PP = P[w_1, w_2, ..., w_N]^{-\frac{1}{N}} \tag{3.12}$$

onde $P[w_1, w_2, ..., w_N]$ é a probabilidade de ocorrência da sequência de palavras $w_1, w_2, ..., w_N$. A título de exemplo vamos considerar um sistema de reconhecimento de dígitos, $(zero, um, dois, ..., nove)$ em que cada um dos 10 dígitos pode ocorrer com igual probabilidade $P = \frac{1}{10}$. Segundo a equação 3.12 a perplexidade deste sistema é 10.

$$\begin{aligned} PP &= P[w_1, w_2, ..., w_N]^{-\frac{1}{N}} \\ &= (\frac{1}{10}^N)^{-\frac{1}{N}} \\ &= \frac{1}{10}^{-1} \\ &= 10 \end{aligned} \tag{3.13}$$

A tabela 3.1 apresenta alguns valores típicos para a perplexidade dos modelos da linguagem, segundo o domínio de aplicação dos respectivos sistemas de reconhecimento.

3.5 Modelos Acústicos

Como já vimos anteriormente, uma realização acústica é representada por uma sequência de vectores extraídos do sinal de fala. **Os modelos acústicos permitem calcular a verosimilhança da sequência de observações dado o modelo de cada uma das realizações acústicas possíveis**. No caso de reconhecimento baseado em palavras, isto corresponde em calcular o valor de $P[O|w_k]$ para cada palavra pertencente ao vocabulário $V : \{w_1, ..., w_K\}$, onde V é o conjunto de palavras

passiveis de serem reconhecidas. **A palavra reconhecida, $\hat{w}$, é aquela cujo modelo maximiza a probabilidade de ter gerado a sequência observada**, isto é:

$$\hat{w} = \max_{w_k} P[O|w_k] \tag{3.14}$$

Em aplicações reais os modelos acústicos não correspondem a palavras, mas sim a fonemas. Os tipos de modelos mais utilizados são os **monofones e os trifones**.

Os monofones não reflectem as diferenças relativas aos efeitos de coarticulação com os fonemas envolventes, pelo que são designados de modelos independentes do contexto. Por sua vez, **os trifones incorporam informação sobre o contexto**. Cada trifone classifica de forma diferenciada a ocorrência de um fonema, de acordo com o fonema imediatamente anterior e do fonema seguinte, também designados por **contexto à esquerda e à direita**, respectivamente.

3.5.1 Modelos de Markov não observáveis

A teoria relativa aos HMM's é bem conhecida e documentada. Assim sendo, neste capítulo, apenas serão apresentados os conceitos básicos e notações importantes para a compreensão dos capítulos posteriores. A teoria relativa aos HMM's pode ser consultada em [29].

Um HMM representa um processo estocástico duplo, com estados internos (não observáveis) e símbolos externos (observáveis). Em reconhecimento de fala pode-se relacionar os estados internos com a **variabilidade temporal** do sinal acústico de fala, enquanto que os símbolos externos representam o **conjunto de possíveis observações** (fonemas) em cada estado [31].

Estrutura dos HMM's

A sequência de estados de um HMM é definida como sendo uma cadeia de Markov discreta, a emissão de símbolos dá-se nas transições de estado. Os elementos que caracterizam um HMM são [29]:

N **-** Número de estados do modelo.

M **-** Número de símbolos que podem ser observados em cada estado. Isto é, o comprimento da sequência de observações.

$S : \{s_1, ..., s_N\}$ **-** Conjunto de estados, incluindo os estados inicial e final.

$A : \{a_{ij}\}$ **-** Matriz fdp de transições entre estados, onde os a_{ij} representam a probabilidade de ocorrer uma transição do estado i para o estado j.

$B : \{b_{jk}\}$ **-** Matriz fdp de emissão de símbolos onde b_{jk} é a probabilidade de ocorrer um símbolo k, quando se atingir o estado j.

$\Pi : \{\pi_j\}$ **-** Matriz de fdp dos estados iniciais, onde π_j é a probabilidade do processo iniciar no estado j.

$O : \{o_1, ..., o_M\}$ **-** Sequência de símbolos observada.

Um modelo λ está completamente definido se as matrizes A, B e Π forem conhecidas. Como estas são matrizes de fdp, têm as seguintes propriedades:

$$
\begin{aligned}
a_{ij} &\geq 0 \quad \forall \quad i, j \\
b_{jk} &\geq 0 \quad \forall \quad j, k \\
\pi_j &\geq 0 \quad \forall \quad j \\
\sum a_{ij} &= 1 \quad \forall \quad i, j \\
\sum b_{jk} &= 1 \quad \forall \quad j, k \\
\sum \pi_j &= 1 \quad \forall \quad j
\end{aligned}
$$

Em reconhecimento de fala é comum utilizar HMM's de primeira ordem. Num HMM de primeira ordem a transição entre estados apenas depende do estado imediatamente anterior ao estado actual, isto é:

$$a_{ij} = P[S_{t+1} = s_j | S_t = s_i] \tag{3.15}$$

Quanto à emissão de símbolos, esta apenas depende do estado actual:

$$b_{jk} = P[O_t = o_k | S_t = s_j] \tag{3.16}$$

A matriz Π é definida da seguinte forma:

$$\pi_j = P[S_1 = s_j] \tag{3.17}$$

Classificação dos HMM's

Os HMM's podem ser classificados de duas formas distintas, tendo em conta a natureza dos elementos da matriz B, e a forma como as transições entre estados podem ocorrer.

Quanto à natureza dos elementos da matriz B, estes podem ser contínuos ou discretos. **Em reconhecimento de fala utilizam-se HMM's discretos**. Nestes, as densidades de probabilidades são definidas em espaços finitos. Neste caso, as observações são vectores de símbolos de um alfabeto finito de M elementos diferentes.

Em relação à forma como as transições entre estados podem ocorrer, o modelo mais geral permite transições entre quaisquer estados (figura 3.6), no entanto existem modelos mais restritivos em que as transições entre estados estão bem definidas. **Em reconhecimento de fala é usual a utilização de modelos do tipo *left-to-rigt***, figura 3.7. Nestes modelos, o processo parte do estado inicial e dirige-se para o estado final sem poder transitar para estados anteriores.

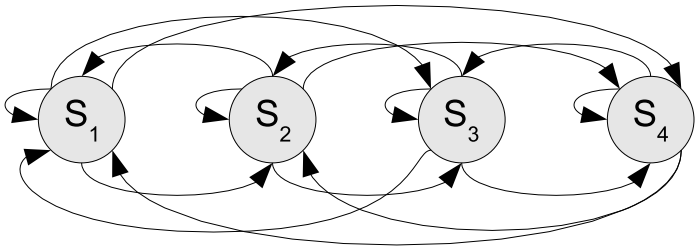


Figura 3.6: HMM ergódico.

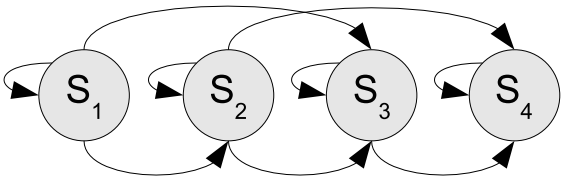


Figura 3.7: HMM do tipo *left-to-rigt*.

A forma como as transições entre estados podem ocorrer depende da aplicação em causa e está intimamente relacionada com a natureza do fenómeno que se pretende modelar.

Treino dos HMM's

A estimação dos parâmetros dos HMM's é feita com recurso a dados de treino e geralmente utilizando o algoritmo *forward-backward* também conhecido como algoritmo de *Baum-Welch*. O critério utilizado para a reestimação dos parâmetros é o de máxima ***Maximum Likelihood*** (ML), que consiste em aumentar, em cada iteração de treino, a probabilidade a posteriori, ou seja, a probabilidade do modelo gerar a sequência de observações [29].

3.5.2 Sistemas de reconhecimento da fala baseados em modelos de Markov não observáveis

Nos sistemas de reconhecimento de fala baseados em modelos de Markov não observáveis, **cada fonema é representado por um HMM de primeira ordem** contendo, tipicamente, três estados e uma topologia do tipo *left-right* como representado na figura 3.8. Os estados de entrada e saída não emissores, são acrescentados ao modelo para facilitar a ligação entre modelos. Em muitos casos, o estado de saída de um modelo é ligado ao estado de entrada de outro, de forma a criar um modelo composto. Isto permite ligar modelos de fonemas, de forma a criar modelos de palavras e com estes criar frases.

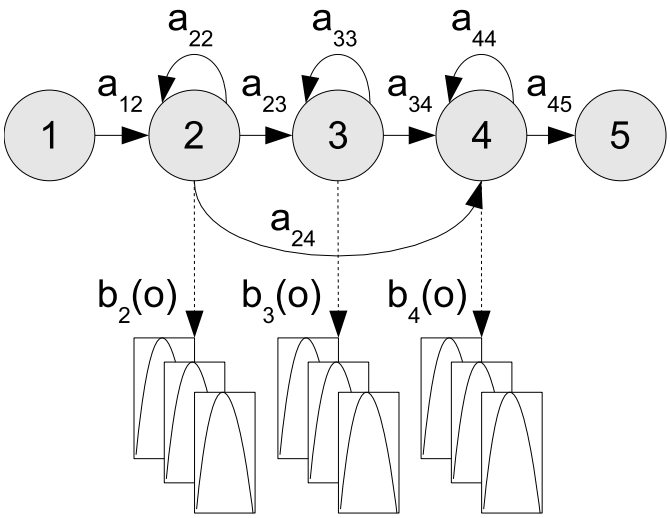


Figura 3.8: HMM de um fonema.

O reconhecimento de fala utilizando HMM's foi introduzido na década de 70 por investigadores

da *Carnegie Mellon University* (CMU) e da *International Business Machines Corporation* (IBM).

3.5.3 Treino de reconhecedores de fala

O treino dos sistemas de reconhecimento de fala é feito a partir de bases de dados com material acústico previamente gravado. Existem diversas bases de dados públicas (para a língua Inglesa) que podem ser utilizadas tanto para treino como para avaliação dos sistemas de reconhecimento. As três mais conhecidas são: **TI Digits, TIMIT e ATIS** [28].

As características mais importantes destas bases de dados são o tipo de discurso e de locutores utilizados na sua criação. Estas características variam tendo em conta o tipo de aplicação a que se destina o sistema de reconhecimento. Quanto ao tipo de discurso, este pode ser: **leitura de fonemas isolados, leitura de palavras isoladas, leitura de frases isoladas, leitura de fragmentos de texto e discurso espontâneo**. A escolha dos locutores envolve vários aspectos. Por exemplo, se o sistema de reconhecimento que se está a desenvolver é para ser utilizado pelo público em geral, então os **locutores devem ser uma amostra representativa dessa população**. Se, pelo contrário, a aplicação é para ser utilizada por uma única pessoa ou por um grupo específico, a escolha dos locutores deve ter em conta estas especificidades [28].

Algumas bases de dados públicas (por exemplo a TIMIT) já têm os dados anotados, isto é, fornecem também a transcrição fonética dos ficheiros de áudio. **A transcrição fonética não é mais que uma correspondência entre os dados acústicos e linguísticos**, ou seja, a transcrição fonética pretende mapear no sinal acústico a informação linguística que este representa.

À data da realização deste trabalho não existiam bases de dados públicas (com dados acústicos) para o português europeu. Assim, foi necessário criar uma com os dados acústicos necessários à realização do reconhecedor. Uma vez que o processo de anotação manual é bastante moroso, optou-se por efectuá-la automaticamente. Ao processo de treino com anotação automática dá-se o nome de *embedded training*, que é realizado recorrendo ao algoritmo de *Baum-Welch* [30].

3.6 Descodificador

Nas secções anteriores foram apresentados os principais componentes de um reconhecedor de fala típico: extracção de parâmetros, modelo da linguagem e modelo acústico. Se recordarmos a figura 3.1 verificamos que falta falar do descodificador. **O descodificador é responsável por determinar qual a sequência de estados de um HMM que tem maior probabilidade de ter gerado a sequência de observações em análise**. Os vectores de observações que é necessário analisar não fornecem indicação clara das fronteiras entre as palavras de uma locução, nem mesmo do número total de palavras que esta contém. A tarefa de determinar o número de palavras presentes numa locução, bem como das fronteiras entre elas, é também uma tarefa a realizar pelo descodificador [30].

Durante o processo de descodificação todos os modelos acústicos são avaliados para calcular a probabilidade de terem gerado um determinado vector de observações. Como o número de modelos aumenta com o vocabulário, podem existir espaços de busca muito alargados, o que torna esta tarefa muito mais lenta que as restantes. Nos sistemas mais desenvolvidos, o processo de descodificação é responsável por praticamente toda a carga computacional requerida pelo sistema. Desta forma, pode-se concluir que é responsável pela velocidade de reconhecimento do sistema [30]. Para descrever o processo de descodificação vamos partir da equação 3.3:

$$\hat{W} = \arg\max_W \{P(O|W)P(W)\}$$

O termo $P(O|W)$ da equação 3.3 tem que ser expandido de acordo com a natureza dos modelos em utilização, neste caso HMM's. **Neste contexto, W não deve ser entendido como um conjunto de palavras, mas sim como um conjunto de modelos acústicos**. Assim sendo, cada um dos modelos acústicos w_k contém uma sequência de **estados internos** $S = \{s_1, ..., s_N\}$. Se fizermos corresponder a sequência de observações $O = \{o_1, ..., o_M\}$ às emissões dos estados internos dos modelos acústicos, o termo $P(O|W)$ pode ser calculado da seguinte forma:

$$P(O|W) = \sum_S P(O, S|W) \tag{3.18}$$

O termo $P(O, S|W)$ representa a probabilidade da sequência de observações O ser produzida pela sequência de estados S (tendo em conta todas as transições possíveis) do modelo em análise.

A escolha da sequência de palavras reconhecida é então [30]:

$$\hat{W} = \arg\max_W \{P(W) \sum_S P(O,S|W)\}$$

O cálculo do somatório $\sum_S P(O,S|W)$, é **bastante dispendioso do ponto de vista computacional**. Nos casos em que o número de palavras (que se pretende reconhecer) é bastante elevado, torna-se impossível fazer os cálculos em tempo útil. Para resolver esta limitação é usual substituir o somatório pela **aproximação de *Viterbi***, equação 3.19 [30] [32].

$$P(O|W) \approx \max_S P(O,S|W) \tag{3.19}$$

Ao invés de considerar todas as transições possíveis entre os estados do modelo em análise, a aproximação de *Viterbi* tem em conta apenas o percurso que leva ao estado mais provável. Ou seja, a transição de estado é feita sempre para aquele cuja probabilidade $P(O,S|W)$ é mais elevada. **Ao processo de eliminação dos caminhos com menor probabilidade dá-se o nome de *pruning*** [30].

Neste processo de maximização, o processo de busca pode ser descrito como uma rede onde se procura o melhor alinhamento temporal entre a sequência de entrada e os estados dos modelos. **Esta procura pode ser feita recorrendo a algoritmos de programação dinâmica**.

3.7 Avaliação

A métrica normalmente utilizada para avaliar sistemas de reconhecimento de fala é a taxa de erro de palavra (***word error rate***) (WER) [30]. A WER é definida da seguinte forma:

$$WER = 100\frac{I+S+E}{T} \tag{3.20}$$

onde I, S, E e T representam **Inserções, Substituições, Eliminações e Total de palavras da transcrição correcta**, respectivamente. O número de inserções, substituições e eliminações é calculado pelo algoritmo de ***minimum edit distance*** [30]. A figura 3.9 apresenta um exemplo da aplicação deste algoritmo, neste caso temos $I = 0$, $S = 1$ e $E = 2$. Supondo que a frase de referência é **"Desligar a Luz da Cozinha"** temos $T = 5$. Neste caso a WER é de 60%.

```
Desligar a luz da cozinha
Ligar      luz    cozinha
------------------------
    S    E      E
```

Figura 3.9: Cálculo da *minimum edit distance* entre duas frases.

3.8 Comentários finais

O problema de reconhecimento de fala é bastante complexo. Neste capítulo abordamos os conceitos mais importantes, os quais, são estritamente necessários à compreensão do trabalho realizado.

Apresentamos os factores que condicionam e dificultam o projecto de reconhecedores de fala: **tamanho do vocabulário, tipo de reconhecedor (dependente ou independente do orador) e a variabilidade do sinal de fala**. Definimos quais os **componentes de um reconhecedor típico** (extracção de parâmetros, modelo da linguagem, modelos acústicos e descodificador), fazendo uma pequena descrição de cada um deles. Por fim, apresentamos a **métrica** mais utilizada para avaliar sistemas de reconhecimento de fala, a *WER*.

Capítulo 4

Desenvolvimento de uma interface *Speech Enabled* para pessoas com limitações funcionais

Com este trabalho **pretendemos desenvolver uma interface baseada em reconhecimento de fala, cuja utilização seja transparente para os seus utilizadores**. Esta aplicação será integrada com o sistema domótico B-LIVE. Desta forma esperamos que pessoas com limitações funcionais graves, (tetra e paraplégicos) possam actuar sobre o B-LIVE, de forma a **aumentarem a sua autonomia e mobilidade dentro de uma habitação**.

Na apresentação desta interface, serão explicados em pormenor todas as etapas de seu desenvolvimento. Começaremos por apresentar o **princípio de funcionamento** da interface desenvolvida. De seguida explicamos a sua **arquitectura**, onde iremos discutir em pormenor os aspectos mais importantes da implementação. O **B-LIVE** será apresentado de uma forma muito sucinta. Apenas serão abordados os assuntos considerados fundamentais para perceber de que forma é que a interface vai comunicar com o B-LIVE. De seguida, apresentamos as características consideradas na **escolha da ferramenta de reconhecimento de fala a utilizar na construção do reconhecedor de fala independente do orador**. Por fim, será descrito o processo de construção dos reconhecedores de fala que iremos utilizar (dependente e independente do orador).

4.1 Princípio de funcionamento

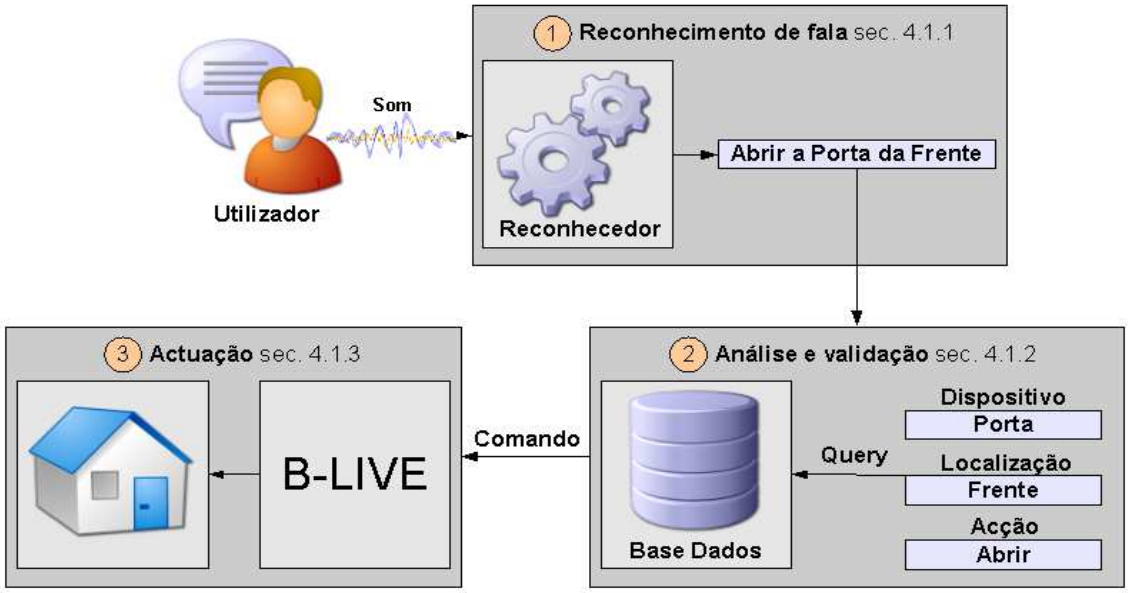


Figura 4.1: Princípio de funcionamento da interface com reconhecimento de fala.

A figura 4.1 apresenta o **funcionamento da interface** desenvolvida para interagir com o sistema domótico B-LIVE. Nela podemos distinguir **três etapas**:

- Reconhecimento de fala
- Análise e validação
- Actuação

4.1.1 Reconhecimento de fala

Esta etapa é responsável por **transformar os sinais acústicos de fala em frases que correspondem às instruções que podem ser executadas pelo sistema domótico B-LIVE**. Sempre que o reconhecedor produz uma frase esta é enviada para o módulo responsável pela **análise e validação**. O reconhecedor de fala será apresentado mais à frente, neste capítulo.

4.1.2 Análise e validação

A análise e validação é responsável por validar as frases vindas do reconhecedor. O processo de análise verifica se a frase **respeita a estrutura previamente definida** (figura 4.2). **A validação verifica se a frase corresponde a uma instrução válida**.

Figura 4.2: Estrutura das frases.

A correspondência entre o conjunto de frases reconhecidas e o conjunto de instruções pode não ser de um para um $(1:1)$, como no exemplo da figura 4.3. Se analisarmos cuidadosamente esta figura, verificamos que a informação relativa aos parâmetros: **Acção, Dispositivo e Localização**, está presente em todas as frases. Neste exemplo, quatro frases diferentes (produzidas pelo reconhecedor de fala) correspondem à mesma instrução ***"Abrir a Porta da Frente"***.

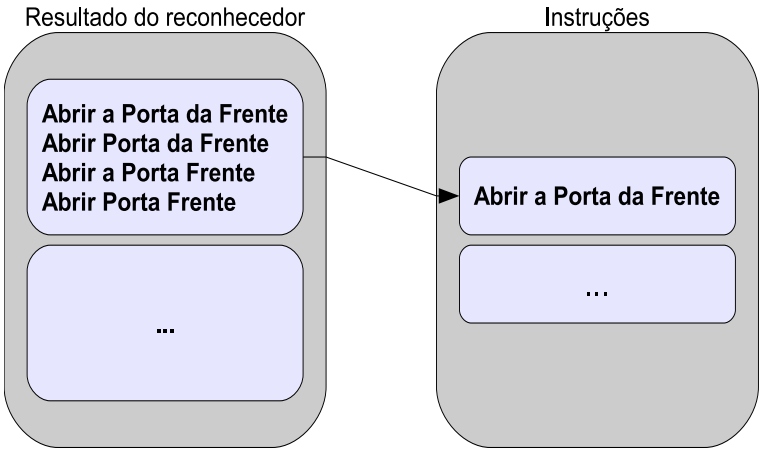


Figura 4.3: Correspondência entre os conjuntos de frases reconhecidas e de instruções.

O que foi dito anteriormente sugere que o processo de análise das frases vindas do reconhecedor passa por uma **identificação dos parâmetros**: Acção, Dispositivo e Localização. **O objectivo é identificar na frase as palavras que os caracterizam**, por exemplo:

Acção — Indica o que se pretende fazer: abrir, fechar, ligar, desligar, subir, descer.

Dispositivo — O dispositivo sobre o qual se vai produzir a acção: porta, estore, lâmpadas, tomada, etc.

Local — Onde se encontra o dispositivo: no quarto, na sala, na frente, na cozinha, etc.

Não é obrigatório que as frases tenham informação relativa a todos os parâmetros, por exemplo, a instrução ***"Autoclismo"*** apenas contém informação àcerca do dispositivo, a acção e local está implícita. Na caso do exemplo que temos vindo a seguir, a figura 4.4 exemplifica o processo de análise da frase ***Abrir a Porta da Frente***.

Frase reconhecida:	Abrir a Porta da Frente
Acção:	Abrir
Dispositivo:	Porta
Local:	Frente

Figura 4.4: Análise da informação contida nas frases.

Agora que já temos a informação relevante contida na frase, para verificar se esta corresponde a uma instrução válida, basta pesquisar a **base de dados**. Esta pesquisa pretende **verificar se existe na base de dados algum comando com a mesma informação** (equação 4.1). Caso exista, o **comando é enviado** para o sistema B-LIVE, caso contrário a frase é **ignorada**.

$$\vec{V}_{FraseReconhecida}(A, D, L) = \vec{V}_{BaseDados}(A, D, L) \tag{4.1}$$

Onde: A representa a acção, D o dispositivo e L o local.

4.1.3 Actuação

Sempre que uma frase é classificada como sendo válida, é necessário executar a tarefa referente à instrução que esta representa. Para isso, apenas é necessário enviar o comando respectivo para o sistema domótico B-LIVE. **A actuação sobre os dispositivos é responsabilidade do B-LIVE**.

4.2 Arquitectura da aplicação de interface

A aplicação de interface, cujo funcionamento foi descrito no ponto anterior, foi desenvolvida segundo uma **arquitectura de camadas** (*Layers*) (figura 4.5). Como se pode ver a partir da figura, esta arquitectura permite ter vários **graus de abstracção**, sendo o mais elevado ao nível da *Interface Layer*.

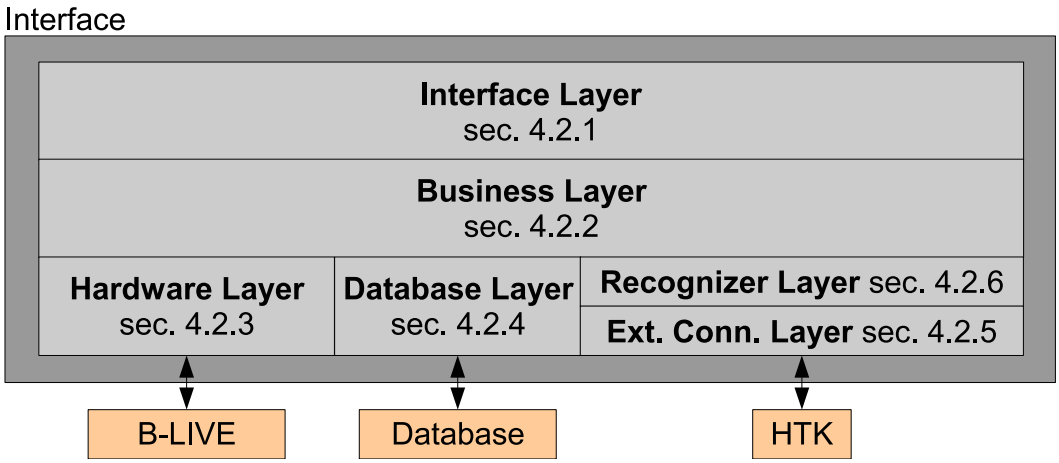


Figura 4.5: Arquitectura da interface *Speech Enabled*.

Cada um dos blocos presentes na figura 4.5 pode ser visto como um bloco de código que disponibiliza métodos para executar determinadas tarefas.

4.2.1 *Interface Layer*

A *Interface Layer* implementa a interface com o utilizador (figura 4.6). As funcionalidades que esta interface apresenta são as seguintes: ligar ou desligar (botões *Live* e *Stop*, respectivamente), o modo de reconhecimento em tempo real e desligar a aplicação (botão *Exit*). Durante o funcionamento apresenta as frases que são reconhecidas e o comando correspondente. **Esta camada de software apenas conhece a existência da *Business Layer*, e é com esta que comunica, de forma a desencadear as acções requeridas pelos utilizadores**.

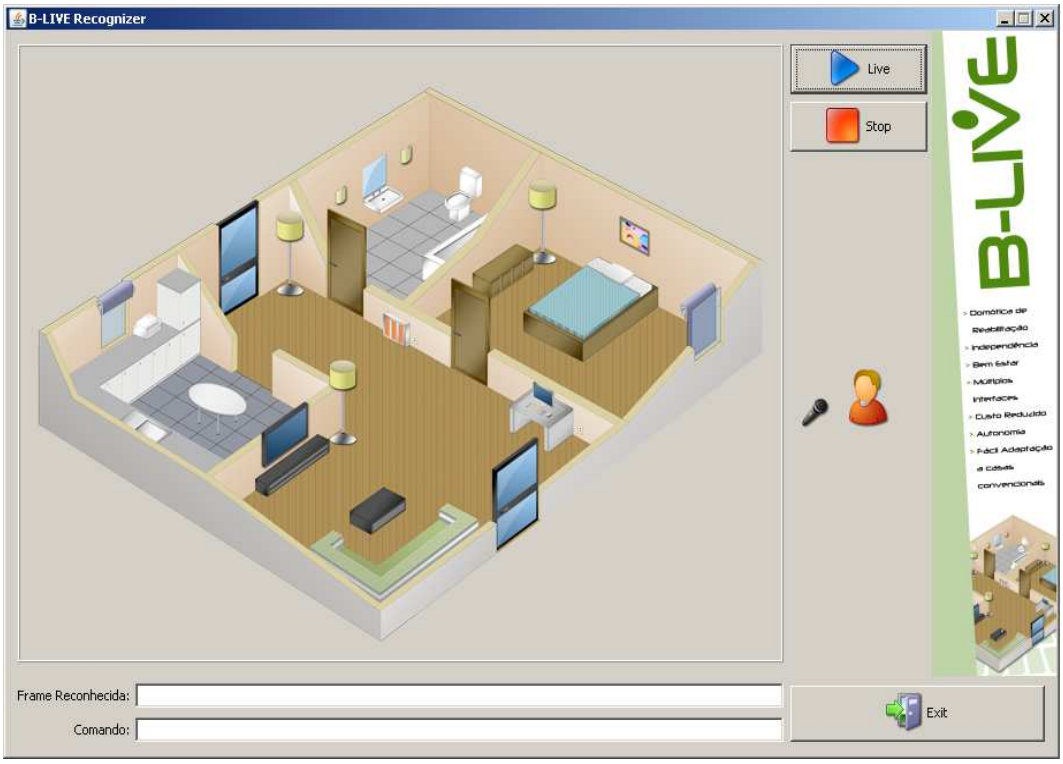


Figura 4.6: Interface com o utilizador. Disponibiliza botões para ligar e desligar o reconhecedor e para sair da aplicação. Apresenta a informação sobre o reconhecimento, vinda do reconhecedor, e o comando correspondente.

4.2.2 *Business Layer*

A *Business Layer* contém toda a lógica da aplicação e é responsável pela comunicação entre os seguintes blocos: *Interface Layer*, *Hardware Layer*, *Database Layer* e *Recognizer Layer*. A figura 4.7 apresenta o **diagrama de classes** da *Business Layer*. Neste diagrama são visíveis as ligações às classes que são utilizadas.

Ligar/Desligar o reconhecedor de fala

Sempre que solicitado pelo utilizador, **a *Business Layer* pode actuar no sentido de ligar ou desligar o reconhecedor de fala**. Para tal apenas tem que evocar os métodos correspondentes da classe ***mioJHVite*** pertencente à *Recognizer Layer*. O funcionamento desta e doutras classes pertencentes à *Recognizer Layer*, será apresentado mais à frente neste capítulo.

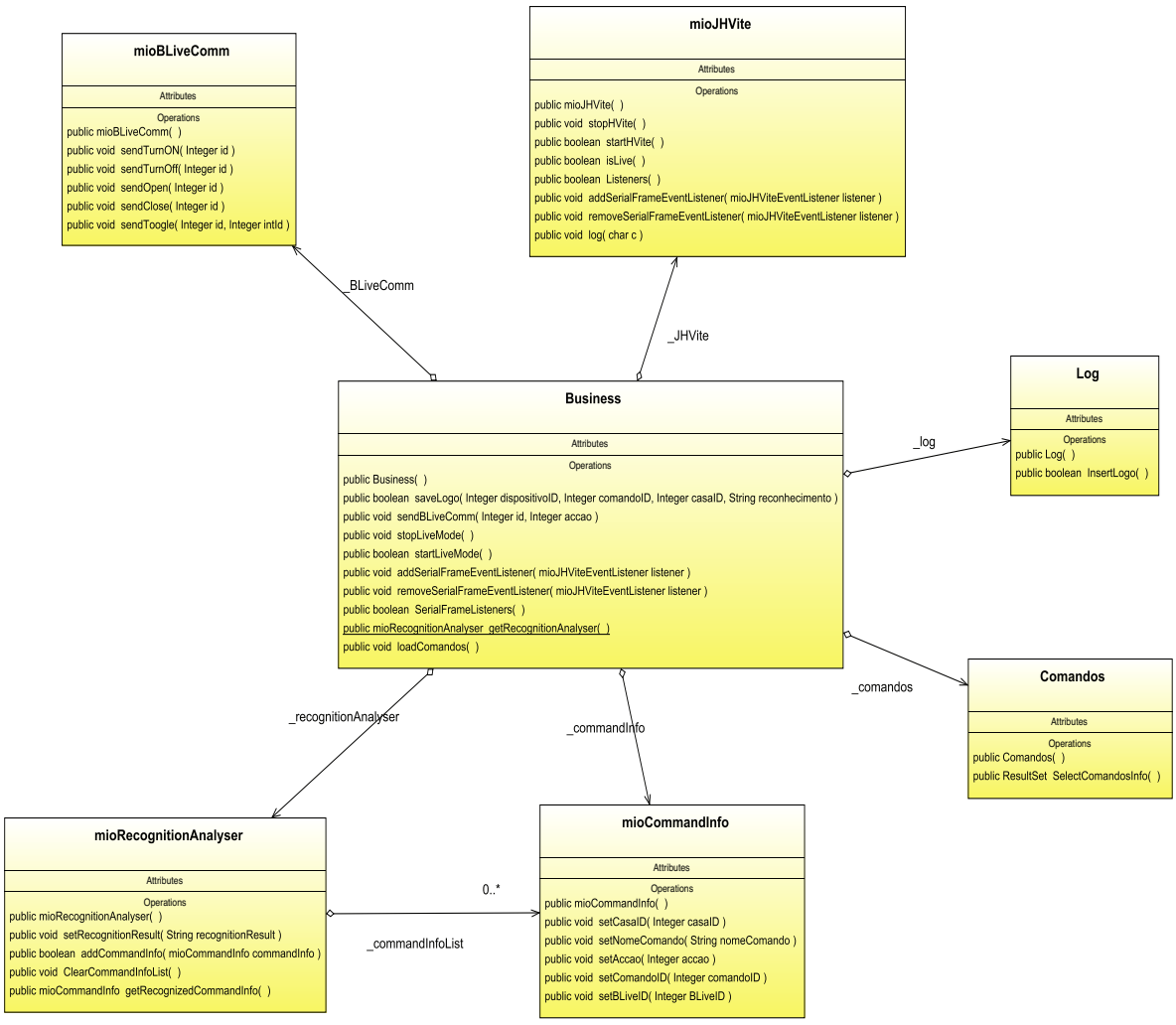


Figura 4.7: Diagrama de classes da *Business Layer*. A *Business Layer* utiliza a classe *mioBLiveComm* para aceder às comunicações, a classe *mioJHVite* para aceder ao reconhecedor de fala, as classes *mioRecognitionAnalyser* e *mioCommandInfo* para analisar as sequências de palavras vindas do reconhecedor e as classes *Log* e *Comandos* para aceder às respectivas tabelas na base de dados.

Análise das frases reconhecidas

É ao nível da *Business Layer* que é feita a análise das frases reconhecidas. Esta tarefa é executada pelo *mioRecognitionAnalyser*. Este analisador necessita de informação àcerca das instruções que podem ser executadas pelo B-LIVE. Esta informação é-lhe fornecida de uma forma estruturada sob a forma de objectos (instâncias da classe ***mioCommandInfo***). Estes objectos são construídos com informação retirada da base de dados recorrendo aos métodos disponibilizados pela classe ***Comandos* da *Database Layer***. Sempre que uma frase é reconhecida, é feito um pedido de análise da mesma ao analisador. Caso a frase corresponda a uma instrução válida o analisador devolve a informação necessária para a executar, caso contrário a frase é ignorada.

Envio de comandos para o B-LIVE

As comunicações com o B-LIVE são implementadas pela classe *mioBLiveComm* pertencente à *Hardware Layer*. Os comandos são enviados para o B-LIVE pela linha série (RS232) através de **mensagens**, encapsuladas em **tramas**. O formato destas tramas será descrito em pormenor mais à frente neste capítulo (na secção: Sistema domótico para pessoas com limitações funcionais: B-LIVE).

Registo das actividades da interface

A aplicação de interface mantém um registo actualizado de todas as acções realizadas. Sempre que uma frase é reconhecida é feito um registo na base de dados do sistema. **Este registo é feito através da classe *Log* da *Database Layer***.

4.2.3 *Hardware Layer*

O acesso ao hardware é disponibilizado pela *Hardware Layer* cujo diagrama de classes está representado na figura 4.8. No caso particular deste trabalho, apenas foi implementado o **acesso à porta série (RS232)**. Para aceder ao driver utilizou-se a livraria ***RXTX*** [33] [34], em cima da qual se desenvolveu o software necessário para comunicar com o B-LIVE.

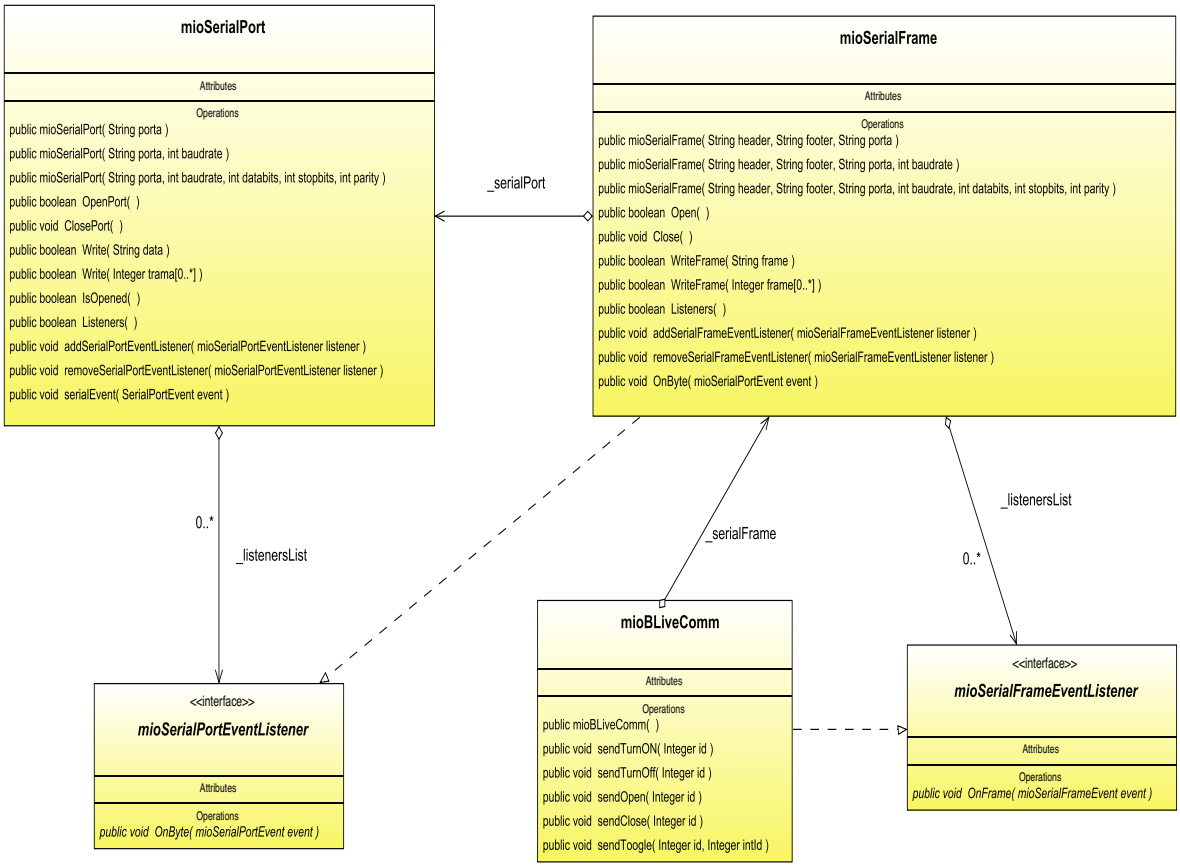


Figura 4.8: Diagrama de classes da *Hardware Layer*. O acesso à porta série é feito pela classe *mioSerialPort*, a classe *mioSerialFrame* é utilizada para detectar e enviar tramas pela porta série, por fim, a classe *mioBLiveComm* é utilizada para enviar comandos.

O **protocolo utilizado para comunicar** com o B-LIVE será apresentado mais à frente, neste capítulo. Por agora, apenas é necessário saber que **as comunicações com o B-LIVE baseiam-se em mensagens**. Estas mensagem são enviadas pela linha série sob a forma de tramas (*frames*).

Acesso à porta série

A classe *mioSerialPort* é responsável pela gestão da porta série (abrir, fechar, ler e escrever na porta), foi desenvolvida para criar alguma abstracção em relação à livraria *RXTX*. Os dados (bytes) recebidos pela porta série chegam à classe ***mioSerialPort*** sob a forma de **eventos** (*serialEvent* despoletado pela livraria *RXTX*) e são disponibilizados da mesma forma.

Os eventos despoletados pela classe *mioSerialPort* são instâncias da classe ***mioSerialPortEvent*** e são enviados para todas as classes que implementem a interface ***mioSerialPortEventListener*** e estejam registadas como receptores. Os métodos necessários à gestão destes eventos são disponibilizados pela classe *mioSerialPort*.

Implementação das *frames*

As *frames* utilizadas nas comunicações com o B-LIVE são implementadas pela classe *mioSerialFrame*. Esta classe utiliza a *mioSerialPort* para aceder à porta série, e implementa a interface *mioSerialPortEventListener* para poder receber os dados que chegam à porta sob a forma de eventos. **As *frames* são delimitadas por um *header* e um *footer* que são utilizados na sua detecção**. Sempre que uma ***frame* é detectada é despoletado um evento**. Estes eventos são instâncias da classe ***mioSerialFrameEvent***.

Os eventos *mioSerialFrameEvent* são enviados para todas as classes que implementem a interface ***mioSerialFrameEventListner*** e estejam registadas como receptores. Os métodos necessários à gestão destes eventos são disponibilizados pela classe *mioSerialFrame*.

Comunicações com o B-LIVE

As comunicações com o B-LIVE são implementadas pela classe *mioBLiveComm*. Esta classe disponibiliza os métodos necessários para **enviar comandos** aos dispositivos controlados pelo B-LIVE, tais como:

- *sendTurnON* — Ligar.
- *sendTurnOff* — Desligar.
- *sendOpen* — Abrir.
- *sendClose* — Fechar.
- *sendTogle* — Alterar estado, por exemplo Ligar para Desligar.

Estes comandos são enviados sob a forma de *frames*, para tal é utilizada a *mioSerialFrame*. Para poder receber *frames* (sob a forma de eventos) vindas do B-LIVE a classe *mioBLiveComm* implementa a interface *mioSerialFrameEventListner* e regista-se como receptor utilizando os métodos disponibilizados pela classe *mioSerialFrame*.

4.2.4 *Database Layer*

A *Database Layer* é responsável pela ligação à base de dados e fornece as classes necessárias para manipulação dos dados nela existentes. A figura 4.9 apresenta o diagrama de classes da *Database Layer*.

Ligação à base de dados

A ligação à base de dados é feita recorrendo à livraria *Java Database Connectivity* (JDBC), fornecida pela *Sun Microsystems, Inc.*. Os dados necessários para efectuar a ligação são fornecidos pelo ficheiro ***conf.props***.

A *mioEngineBase* é uma classe abstracta que define quais os métodos que são indispensáveis para gerir a ligação e aceder aos dados. Esta classe é a base sobre a qual são desenvolvidas as classes dependentes do motor de base de dados. No caso concreto deste trabalho apenas foi desenvolvida uma classe, a ***mioAccess***, para ligar a bases de dados ***Microsoft Access***.

A classe *mioDataBaseServer* é utilizada para criar alguma abstracção em relação ao motor de base de dados que se vai utilizar. Esta classe lê o conteúdo do ficheiro *conf.props* e disponibiliza uma ligação tendo em conta as **configurações nele existentes**.

Acesso aos dados

O acesso aos dados é feito recorrendo a *Stored Queries*, guardadas na própria base de dados. Do lado da *Database Layer*, o acesso às *Stored Queries* está organizado por classes, tendo em conta as tabelas a que se destinam. Assim sendo, existem as seguintes classes: **Casa,**

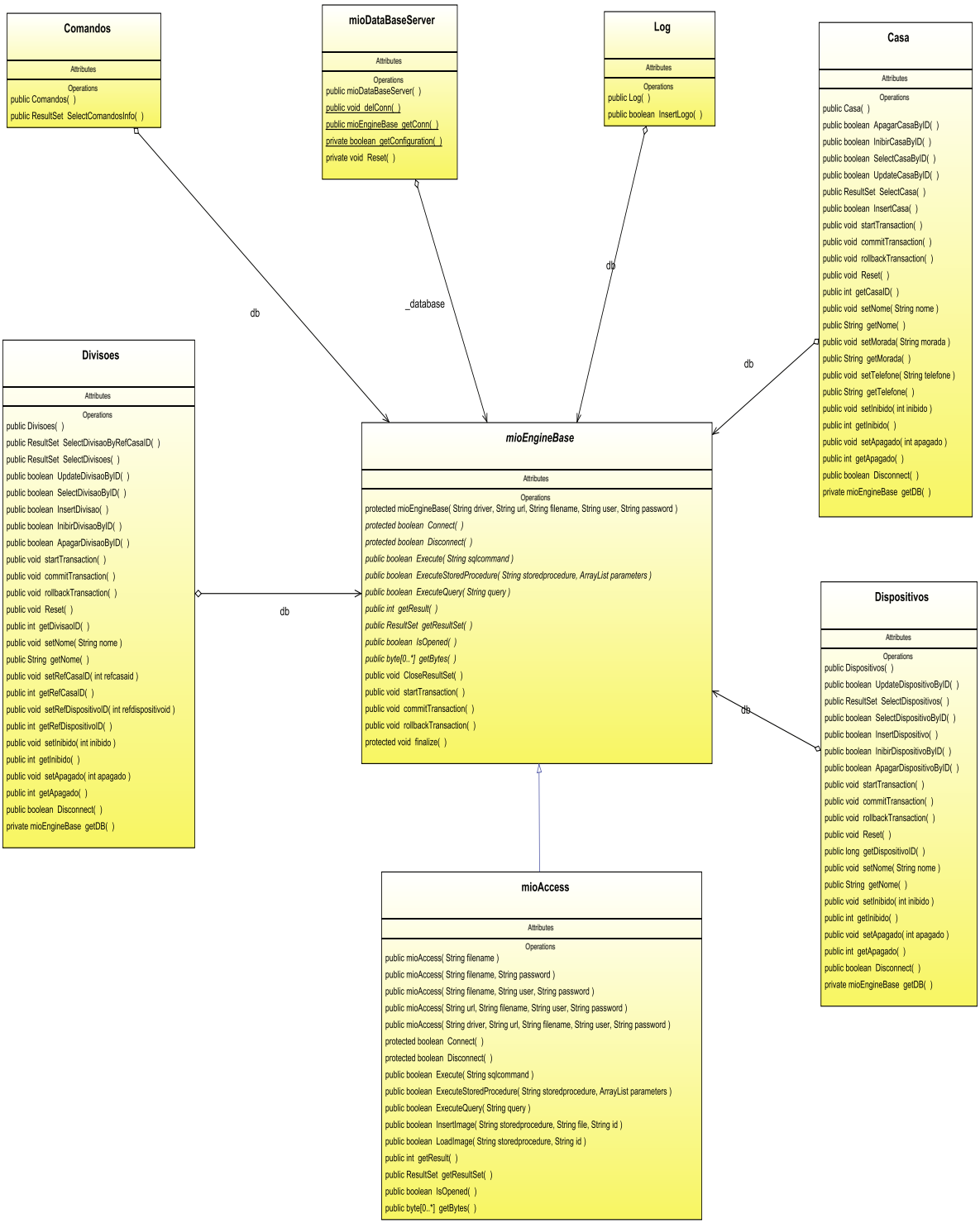


Figura 4.9: Diagrama de classes da Database Layer. A classe *mioAccess* herda a classe *mioEngineBase*. As restantes classes utilizam a *mioEngineBase* para acederem à base de dados.

Divisoes, Dispositivos, Comandos e Log. Estas classes disponibilizam métodos para **consultar, inserir e apagar** informação existente na base de dados.

4.2.5 *External Connection Layer*

A *External Connection Layer* faz a ligação entre a aplicação de interface (escrita em Java) e blocos de código externos, neste caso com o reconhecedor de fala (escrito em C).

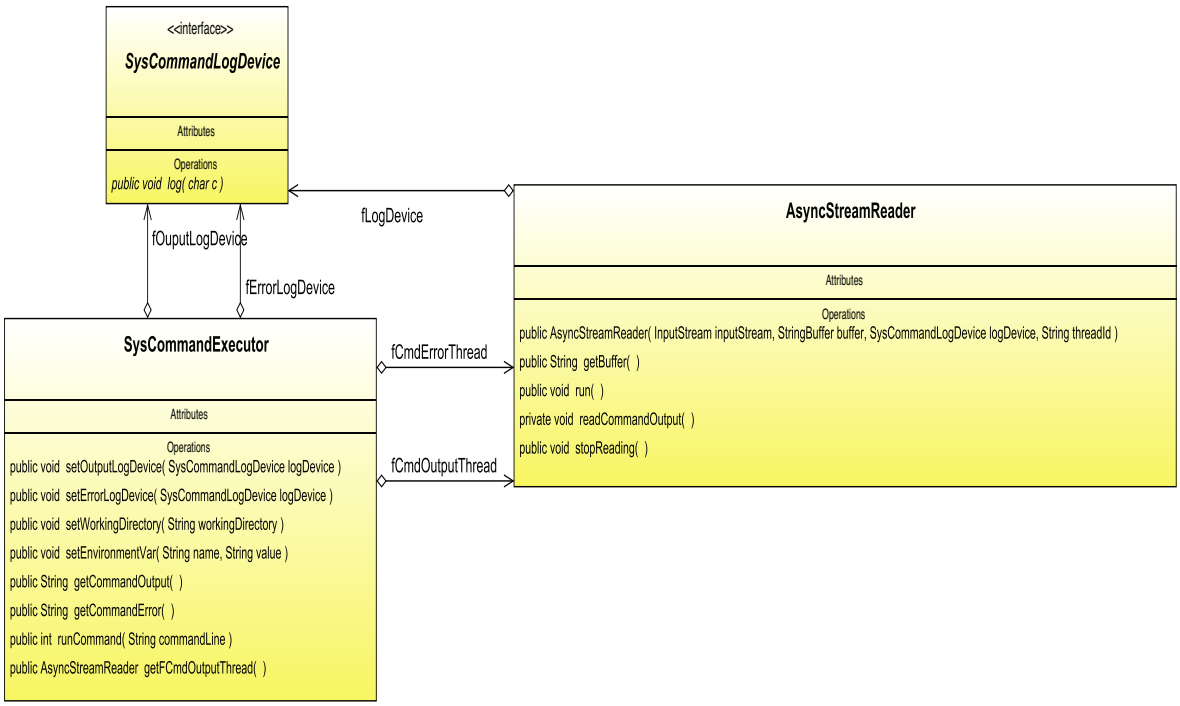


Figura 4.10: Diagrama de classes da *External Connection Layer*.

No caso concreto deste trabalho, o que se pretende é lançar o reconhecedor e recolher os conjuntos de palavras que vão sendo reconhecidos. Uma vez que o reconhecedor corre em modo consola e envia as sequências de palavras que reconhece para o ***Standard Output*** (monitor), podemos utilizar as classes apresentadas no diagrama da figura 4.10 para realizar esta tarefa.

A classe ***SysCommandExecutor*** utiliza classes nativas da linguagem Java para executar comandos externos (neste caso corre um ficheiro *.bat) e redirecciona o seu ***output*** para a classe ***AsyncStreamReader***. Por sua vez, a classe *AsyncStreamReader* lança um evento sempre que

recebe um caracter. Para receber estes eventos noutra classe apenas é necessário implementar a interface ***SysCommandLogDevice***. Estas classes foram retiradas de [35].

4.2.6 *Recognizer Layer*

A *Recognizer Layer* representa uma camada de abstracção em relação à *External Connection Layer* e ao reconhecedor. O seu diagrama de classes é apresentado na figura 4.11.

Figura 4.11: Diagrama de classes da *Recognizer Layer*.

O que se pretende é desenvolver classes que disponibilizem um mecanismo, de fácil utilização, para ligar e desligar o reconhecedor e também para receber as sequências de palavras reconhecidas. A classe responsável por executar estas tarefas é a ***mioJHVite***, a qual disponibiliza métodos para realizar as seguintes operações sobre o reconhecedor: **ligar, desligar e verificar se está ligado**. Estas tarefas são realizadas recorrendo às classes disponibilizadas pela

External Connection Layer. Como vimos anteriormente, a *External Connection Layer* disponibiliza o resultado dos comandos que executa sob a forma de eventos. Para receber estes eventos a classe *mioJHVite* implementa a interface ***SysCommandLogDevice***.

O resultado do reconhecimento chega à classe *mioJHVite* caracter a caracter, pelo que é necessário desenvolver uma estratégia para agrupar os caracteres pertencentes à mesma frase. Para resolver este problema foi introduzido o conceito de *frame*.

Do lado do reconhecedor foi alterada a forma como as frases reconhecidas são enviadas para o *Standard Output*. O objectivo é que as frases reconhecidas sejam delimitadas por um *header* e um *footer* de forma a criar uma *frame*. Desta forma, a classe *mioJHVite* apenas tem que fazer a detecção de *frames* para agrupar os caracteres pertencentes à mesma frase.

Sempre que uma frase é detectada pela classe *mioJHVite*, é lançado um evento *mioJHViteEvent* que contém a frase reconhecida. Para poder receber as frases (sob a forma de eventos) detectadas pela classe *mioJHVite*, é necessário implementar a interface ***mioJHViteEventListner*** e registar-se como receptor utilizando os métodos disponibilizados pela classe *mioJHVite*.

4.3 Base de Dados

Nas secções anteriores descreveu-se o princípio de funcionamento e a arquitectura da aplicação de interface, desenvolvida para interagir com o sistema domótico B-LIVE. Contudo, **a interface de nada serve se não tiver informação sobre o sistema**. **Esta informação está estruturada numa base de dados relacional, desenvolvida com a ferramenta *MS Access***.

Existem várias ferramentas para desenvolver bases de dados relacionais em sistemas baseados em PC, tais como: ***SQL Server*, *MySQL*, *PostgreSQL* ou *MS Access***. No processo de selecção da ferramenta a utilizar tiveram-se em conta os seguintes aspectos: **quais as necessidades do sistema, o licenciamento e a facilidade de instalação**. A escolha recaiu sobre o ***MS Access*** essencialmente pelo facto de não requerer instalação, basta copiar o ficheiro que contém a base de dados.

A informação existente na base de dados está estruturada num conjunto de tabelas, da

seguinte forma:

tblCasa — **Esta tabela contém a informação considerada relevante sobre a casa onde o B-LIVE está instalado**: Nome (nome da casa), Morada (local onde se encontra a casa), Telefone (número de telefone, caso exista).

tblDivisoes — **Contém informação sobre as divisões existentes na casa**: Nome (nome da divisão), RefCasaID (referência para a casa que contém a divisão), RefDispositivoID (referência para um dispositivo que pertencente a esta divisão).

tblDispositivos — **Define quais os dispositivos sobre os quais o sistema B-LIVE pode operar**: Nome (nome do dispositivo).

tblComandos — **Contém os comandos relativos às acções que se podem efectuar sobre os diversos dispositivos, bem como informação sobre a qual dispositivo corresponde um comando**: Comando (nome do comando), RefDispositivoID (referência para o dispositivo sobre o qual se vai executar o comando), Accao (acção que se pretende executar: A - Alterar o estado, O - Abrir, Ligar ou Subir e F - Fechar, Desligar ou Descer).

tblLogo — **Mantém um registo de tudo o que se passa no sistema**: RefDispositivoID (referência para o dispositivo), RefComandoID (referência para o comando), RefCasaID (referência para a casa), Reconhecimento (frase reconhecida), AudioPath (caminho para o ficheiro de áudio que contém a ordem pretendida), Data (data da ocorrência da acção), Hora (hora em que ocorreu a acção).

Os campos "CasaID", "DivisaoID", "DispositivoID", "ComandoID" e "LogID" são as chaves primárias das respectivas tabelas. A chave primária de uma tabela **nunca se repete**, pelo que pode ser utilizada como **referência** para indexar os registos da tabela. As relações entre as tabelas existentes na base de dados estão ilustradas na figura 4.12.

Os campos Inibido e Apagado presentes em todas as tabelas, classificam os registos (linhas da tabela) quanto à sua validade. Se o campo **"Inibido"** for falso significa que o registo está activo (pode ser utilizado), caso contrário deve ser ignorado. Quanto ao campo **"Apagado"**, este indica que o registo nunca mais será utilizado. Em alternativa poderia-se eliminar o registo. Não o fazemos porque perderíamos informação sobre os **acontecimentos passados**.

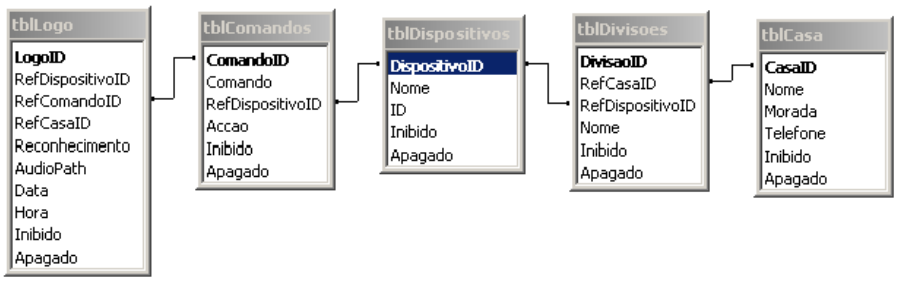


Figura 4.12: Relações entre as tabelas da base de dados.

As pesquisas à base de dados são feitas com recurso a *Stored Queries*. Desta forma agiliza-se o processo de pesquisa uma vez que estas estão **pré compiladas** no ficheiro da base de dados. Outra vantagem é a **independência entre a aplicação e a base de dados**, ou seja, podem ser adicionadas ou alteradas *Stored Queries* sem que seja necessário proceder a grandes alterações do lado da aplicação.

4.4 Sistema domótico para pessoas com limitações funcionais: B-LIVE

Este capítulo foi escrito tendo por base a **documentação técnica do sistema B-LIVE**, da qual o autor desta dissertação foi um dos responsáveis.

Neste capítulo será apresentado o sistema de domótica para pessoas com limitações funcionais, B-LIVE. **Este sistema tem como objectivos melhorar as condições funcionais de habitabilidade de casas convencionais para doentes com mobilidade extremamente reduzida**, nomeadamente, tornando-os suficientemente autónomos para que seja dispensada a presença em permanência de um acompanhante. O sistema tem em consideração que a tecnologia a introduzir tem de ser de custo controlado para que a sua aplicação seja viável.

Será efectuada uma descrição da arquitectura do sistema, do protocolo de comunicação, da arquitectura do software e das várias *Human-Machine Interfaces* (HMIs).

4.4.1 Arquitectura do sistema B-LIVE

Uma das principais características do B-LIVE é a modularidade, figura 4.13. **Cada um dos módulos que o constitui pode funcionar *per si*, ou em conjunto com os restantes**. Para o efeito existe uma **rede de comunicação entre todos os módulos do sistema**. A introdução de novas funcionalidades no sistema é, assim, reduzida à simples operação de introduzir um ou mais módulos.

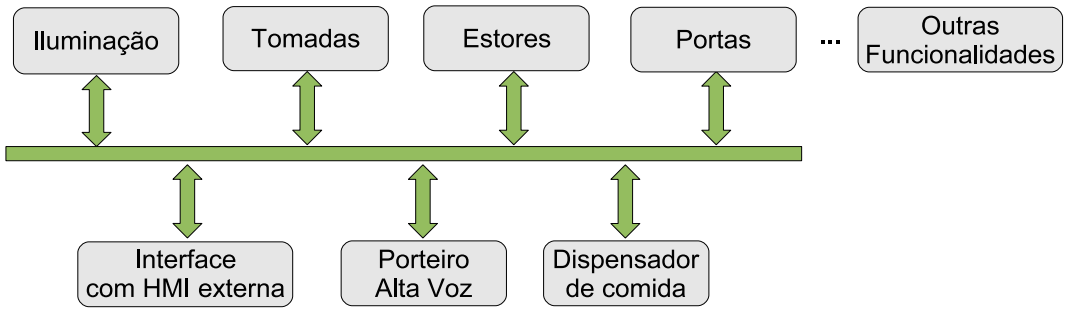


Figura 4.13: Arquitectura do sistema B-LIVE.

Como se pode ver na figura 4.14, o sistema dispõe de **várias interfaces** para comunicar com o exterior, as quais podem funcionar em simultâneo. Do ponto de vista das funcionalidades, o sistema pode receber comandos provenientes de várias interfaces distintas, HMI's ou não (por exemplo, um alarme lançado por um sensor).

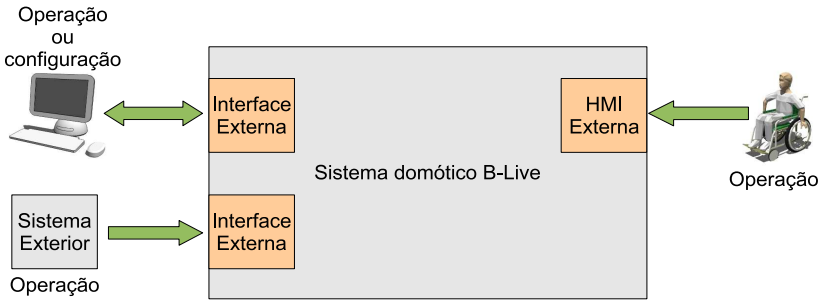


Figura 4.14: Interacção do sistema B-LIVE com o exterior.

Arquitectura dos módulos

Como foi referido anteriormente, o sistema B-LIVE é composto por módulos. As figuras 4.15 e 4.16 representam a arquitectura de cada um dos módulos.

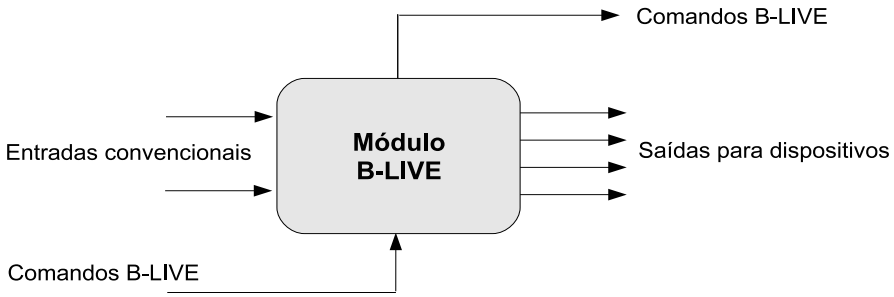


Figura 4.15: Arquitectura dos módulos B-LIVE.

Cada módulo tem **duas entradas** convencionais, para actuadores manuais como interruptores, e **quatro saídas**, para actuar sobre os diferentes dispositivos que estão a ser controlados pelo módulo. Os módulos podem também **receber e/ou enviar comandos através de um barramento**. **A customização dos módulos depende apenas da placa de potência ou adaptação especializada**.

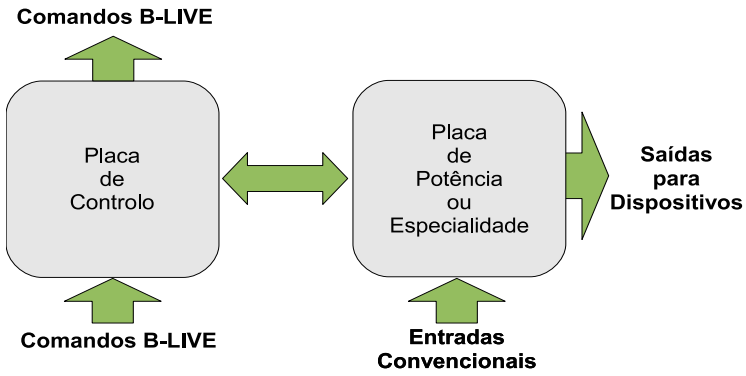


Figura 4.16: Arquitectura interna dos módulos B-LIVE.

4.4.2 Protocolo de comunicações utilizado pelo B-LIVE

Como foi referido na arquitectura do sistema B-LIVE, **existe uma rede de comunicação que interliga todos os módulos**. Presentemente, essa rede é constituída pelo *fieldbus* ***Controller Area***

***Network* (CAN)**. Este protocolo foi desenvolvido pela Bosch na década de 80 e é largamente utilizado na indústria automóvel. **Para comunicar com o exterior o B-LIVE utiliza a linha série (RS232)**.

As comunicações internas (entre os módulos) não são relevantes no âmbito deste trabalho, pelo que apenas vamos descrever o protocolo utilizado nas comunicações feitas através da linha série.

Linha série (RS232)

A Figura 4.17 representa a **estrutura das *frames*** trocadas pela linha série.

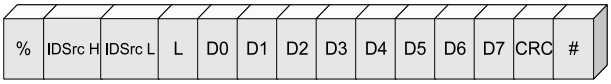


Figura 4.17: Estrutura das *frames* trocadas pela linha série.

Cada um dos campos da *frame* tem 8 bits. O significado dos campos é o seguinte:

- *%* - Inicializador da *frame*.
- *IDSrcH* - Identificação do módulo (3 bits menos significativos).
- *IDSrcL* - Identificação do módulo.
- *L* - Tamanho do campo de dados.
- *D0-D7* - Dados contidos na *frame*.
- *CRC* - Checksum.
- *#* - Terminador da *frame*.

4.4.3 *Firmware* B-LIVE

O *firmware* B-LIVE é comum a todos os módulos. As funcionalidades que cada módulo adquire são definidas posteriormente através do processo de configuração. De seguida será efectuada uma breve descrição da arquitectura do *firmware* B-LIVE.

4.4.4 Arquitectura do *Firmware* B-LIVE

A arquitectura do software dos módulos B-LIVE encontra-se dividida em três componentes, figura 4.18): ***Application*, *Drivers* e *Hardware Interfaces***. Dependendo das funcionalidades dos módulos, podem existir diferentes *drivers* que estabelecem a ligação entre as componentes *Application* e *Hardware Interfaces*. A *Application* executa diferentes gestores de modo a executar diversas tarefas: armazenar a configuração (*Config Manager*) e estados (*Status Manager*) dos módulos, definir as funções específicas dos módulos (*Specific Function Manager*), gerir as comunicações locais (*Local Communication Manager*) e remotas (*Remote Communication Manager*) e as saídas/entradas digitais (*Digital I/O Manager*).

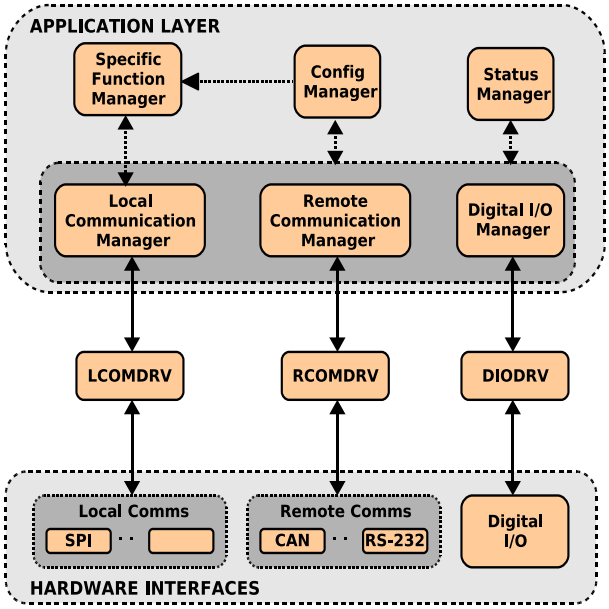


Figura 4.18: Arquitectura do *firmware* B-LIVE [8].

O motivo que levou à escolha desta arquitectura deve-se ao facto de os módulos poderem utilizar vários protocolos de comunicação locais (SPI, I2C, etc.) e remotos (CAN, ZigBee, Ethernet, etc.) em simultâneo. Deste modo, um driver (LCOMDRV) é responsável pela gestão de todas as comunicações locais, enquanto o outro driver (RCOMDRV) faz a gestão das comunicações remotas. Esta opção, comparando com a utilização um driver para cada um dos protocolos de comunicação, permite reduzir a complexidade da camada de drivers e simplifica a coordenação entre as comunicações locais ou remotas. Deste modo é possível escalonar comunicações locais e remotas de acordo com os parâmetros especificados pelo utilizador/aplicação (por exemplo, priori-

dade).

4.4.5 Operação do sistema B-LIVE

Um dos aspectos mais importantes no desenvolvimento de sistemas de domótica para pessoas com limitações funcionais são as HMI's [36]. De facto, não basta desenvolver sistemas de baixo custo, robustos, fiáveis e com diversas funcionalidades. É necessário que o utilizador final disponha de **interfaces intuitivas e de fácil utilização** para operar sobre os sistemas [37] e [38].

Uma das principais características do B-LIVE é a modularidade e facilidade de adaptação às necessidades dos utilizadores. Tendo em conta as necessidades específicas de cada utilizador podem ser utilizadas **várias interfaces para interagir com o B-LIVE**. Actualmente, as interfaces disponíveis para o sistema B-LIVE são as seguintes:

- Computador.
- Telemóvel.
- Interface feita por medida.
- Interruptores de boca (IntegraMouse e IntegraSwitch [39]).
- Interruptores convencionais.

IntegraMouse* e *IntegraSwitch

O ***IntegraMouse*** e o ***IntegraSwitch*** apresentados na figura 4.19, são interfaces Homem-máquina para pessoas com graves limitações funcionais.

O *IntegraMouse* apresenta as mesmas funcionalidades que um rato para computador convencional. A principal diferença é que este é operado através da boca e dos lábios para efectuar movimento e o sopro ou sucção são utilizados para simular os botões. O utilizador com limitações pode recorrer ao *IntegraMouse* para operar o menu do computador referido anteriormente.

Figura 4.19: Interface interruptor e rato de boca.

Quando as limitações dos utilizadores não permitem o recurso interruptores adaptados, o *IntegraSwitch* pode ser utilizado em substituição. O sentido do ar (sopro ou sucção) no *IntegraSwitch* simula os botões de um interruptor.

4.4.6 Conclusão

Nesta secção foi apresentado (de uma forma muito simplificada) o B-LIVE, um sistema de domótica para pessoas com limitações funcionais, de **baixo custo**, **descentralizado** e que utiliza técnicas de ***retrofitting***. Foram descritos a arquitectura do sistema, o protocolo de comunicação, a arquitectura do software e as múltiplas HMI's.

4.5 Selecção da ferramenta de reconhecimento de fala, para construir o reconhecedor dependente do orador

O objectivo deste trabalho não é construir uma ferramenta de reconhecimento de fala de raiz, mas sim escolher uma existente e utilizá-la para construir um reconhecedor. O processo de escolha vai-se limitar a ferramentas que não acarretem custos na sua aquisição, isto é, apenas vão ser estudadas **ferramentas de domínio publico** que possam ser utilizadas juntamente com aplicações comerciais.

Este estudo vai-se restringir apenas às duas ferramentas mais utilizadas em reconhecimento de fala. A primeira foi desenvolvida pela *Cambridge University* e designa-se ***Hidden Markov Model Toolkit* (HTK)**. A segunda é financiada pela *Defense Advanced Research Projects Agency* (DARPA) e foi desenvolvida pela *Carnegie Mellon University* (CMU). Esta é conhecida como projecto ***Sphinx***.

Como em qualquer outro processo de escolha, é necessário definir quais os critérios de avaliação que vão ser utilizados. No caso particular deste trabalho, o que se pretende é uma ferramenta de reconhecimento que cumpra, de forma satisfatória, a maior parte dos seguintes itens:

Acesso ao código fonte — O acesso ao código fonte permite **maior flexibilidade** na construção de aplicações, particularmente quando se pretende fazer integração. Do ponto de vista académico, este acesso é importante para perceber o que se está a passar em cada etapa do processo de reconhecimento.

Portabilidade — É conveniente que o reconhecedor possa ser utilizado em **diferentes sistemas operativos**, desta forma, facilita-se a sua disseminação. No caso especifico deste trabalho, este não é um ponto muito crítico, uma vez que não é significativa a diversidade de sistemas operativos com os quais se vai trabalhar.

Independência do orador — É importante que o reconhecedor possa ser utilizado, com sucesso, por **mais que um utilizador**.

Flexibilidade na escolha do modelo de linguagem — O modelo de linguagem utilizado num reconhecedor pode afectar de forma significativa o seu desempenho. É fundamental testar vários para depois escolher aquele que apresenta melhores resultados.

Possibilidade de introduzir regras linguísticas — A introdução de regras linguísticas é bastante importante. Quando utilizadas correctamente, podem aumentar de forma considerável o desempenho do reconhecedor.

Existência de modelos acústicos para o Português ou forma de os criar — A existência de modelos acústicos, neste caso, para o Português, obtidos a partir de um corpus de grande dimensão, facilita o processo de treino. Quando os modelos acústicos são obtidos a partir de um corpus adequado ao vocabulário que se pretende reconhecer, o desempenho do reconhecedor pode melhorar significativamente.

Possibilidade de treinar o reconhecedor ao longo do tempo — Os modelos utilizados pelo reconhecedor devem poder ser **treinados pelo utilizador final**. Desta forma, espera-se aumentar a taxa de sucesso no reconhecimento, mas também, **adaptar o reconhecedor** a novos utilizadores sempre que seja necessário.

Tipo de Licença — A licença deve ser o mais permissiva possível; deve permitir a utilização e distribuição do software, mesmo em aplicações comerciais.

Informação disponível — A quantidade e qualidade de tutoriais ou demos, é também um factor a ter em conta.

De seguida, vão ser analisadas as ferramentas HTK e *Sphinx*. O estudo será orientado de forma a perceber qual delas é a mais adequada à realização deste trabalho. Questões relacionadas com o funcionamento dos módulos que as constituem não serão abordadas. Aquando da construção do reconhecedor será feito um estudo mais exaustivo sobre o funcionamento e características da ferramenta previamente escolhida.

4.5.1 *Hidden Markov Model Toolkit*

A primeira versão desta ferramenta, o HTK, foi desenvolvida no *Speech Vision and Robotics Group of the Cambridge University Engineering Department* (CUED) em 1989 por Steve Young. **O HTK é um conjunto de livrarias e ferramentas utilizadas para fazer reconhecimento automático de fala usando modelos de Markov não observáveis**. A figura 4.20 ilustra a sua arquitectura.

Este software foi comercializado durante alguns anos pela *Cambridge University* (CU). Em 1999, a Microsoft comprou a empresa que detinha os direitos sobre o HTK, e em 2000 passou a ser **disponibilizado sem custos para o utilizador**. Com esta medida, a Microsoft pretende promover o desenvolvimento e massificar a utilização de reconhecimento de fala em sistemas computacionais.

De seguida analisar-se-á cada uma das características enumeradas no ponto anterior. Não será um estudo exaustivo nem muito pormenorizado. **O que se pretende é perceber se o software se adequa às necessidades deste trabalho**. Esta primeira análise à ferramenta é também bastante importante para perceber quais as etapas a percorrer e quais as dificuldades que podem ocorrer na construção do reconhecedor.

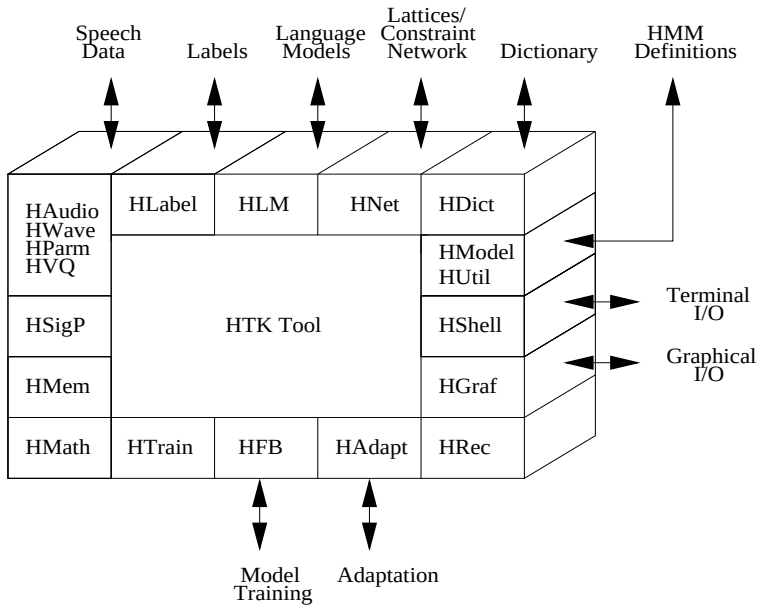


Figura 4.20: Arquitectura da *Hidden Markov Model Toolkit* [40].

Acesso ao código fonte

Os investigadores envolvidos no desenvolvimento do HTK disponibilizam o seu código fonte, o que permite fazer alterações no código de acordo com as necessidades do reconhecedor que se pretende construir.

Portabilidade

A ferramenta HTK é inteiramente desenvolvida em ANSI C. Geralmente é utilizada em sistemas UNIX, contudo, após ser compilada de forma conveniente, pode correr em qualquer sistema operativo.

Independência do orador

Com o HTK é possível construir reconhecedores do tipo dependente ou independente do orador. No caso de se pretender construir um reconhecedor do tipo independente do orador, basta obter o material de treino adequado.

Flexibilidade na escolha do modelo de linguagem

É possível utilizar modelos de linguagem do tipo *statistical N-Gram*, o que permite construir modelos em que o reconhecimento pode ser dependente do contexto. Embora não exista muita flexibilidade a este nível, as ferramentas e documentação disponibilizadas para criar estes modelos são de boa qualidade.

Possibilidade de introduzir regras linguísticas

Existe a possibilidade de introduzir regras linguísticas, por exemplo, anotações. (Não foi possível obter muita informação àcerca deste assunto.)

Possibilidade de treinar o reconhecedor ao longo do tempo

O HTK permite que os modelos de um reconhecedor sejam treinados ao longo do tempo. Este é um ponto importante. Assim, o reconhecedor pode crescer de acordo com as necessidades que vão surgindo ao longo da sua utilização.

Existência de modelos acústicos para o Português ou forma de os criar

Com o HTK não são distribuídos modelos acústicos para o Português. No entanto, estes podem ser obtidos e treinados com o HTK, desde que exista material de treino apropriado.

Tipo de licença

Esta ferramenta está **disponível gratuitamente** mediante o cumprimento das seguintes disposições:

- Pode ser usado para ensino e investigação académica sem restrições.
- Pode ser utilizado por empresas para desenvolvimento de novos produtos.
- Não pode ser incluído, no seu todo ou em parte, em software comercial.

Contudo, **é possível desenvolver aplicações comerciais com base no HTK**. O que não se pode é, juntar as aplicações no mesmo pacote de instalação.

Informação disponível

Quanto à documentação existente, esta é de boa qualidade e bem estruturada sob a forma de livro, o ***HTK Book***. Existem ainda vários tutoriais disponíveis na Internet que explicam como construir reconhecedores utilizando esta ferramenta.

Resumo

A ferramenta HTK é bastante utilizada em laboratórios de todo o Mundo, pelo que está **suficientemente testada** e os resultados obtidos são ao nível dos melhores. **Esta ferramenta tem a vantagem de fornecer todas as aplicações necessárias à construção de um reconhecedor**. **A documentação é de boa qualidade e bem estruturada**. O único senão é a portabilidade, contudo, tendo em conta os possíveis utilizadores do reconhecedor a realizar com este trabalho, este factor não é limitativo, uma vez que a diversidade de sistemas operativos que se podem encontrar não é significativa.

4.5.2 Projecto *Sphinx*

O projecto *Sphinx* começou a ser desenvolvido pela CMU, com o financiamento da DARPA. Ao longo dos últimos anos, foram desenvolvidas várias versões deste software tendo em vista aplicações diferentes. O ***Sphinx-2* é apropriado para reconhecimento em tempo real**, ou para máquinas com pouca capacidade de processamento, contudo, a sua taxa de sucesso no reconhecimento é inferior à da *Sphinx-3* e da *Sphinx-4*. **Os *Sphinx-3* e *Sphinx-4* têm capacidade de efectuar reconhecimento em grandes vocabulários com taxas de sucesso comparáveis**. O ***Sphinx-3* é escrito em C**, enquanto que o ***Sphinx-4* é escrito em Java** e devido à sua arquitectura modular (figura 4.21), é bastante versátil e flexível.

O *Sphinx-4* é uma das mais avançadas ferramentas para reconhecimento de fala. Foi

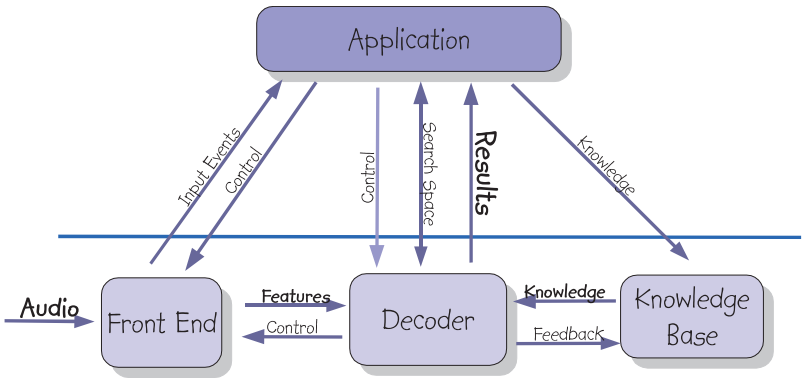


Figura 4.21: Arquitectura do *Sphinx-4* [41].

desenvolvida numa parceria entre a CMU, os *Sun Microsystems Laboratories* (SML), os *Mitsubishi Electric Research Laboratories* (MERL) e a *Hewlett Packard* (HP) com contribuições da *University of California at Santa Cruz* (UCSC) e do *Massachusetts Institute of Technology* (MIT) [42].

Tal como no caso do HTK, é necessário analisar esta ferramenta. As características sujeitas a análise são as mesmas, assim como os propósitos a que esta se destina.

Acesso ao código fonte

O código fonte do *Sphinx-4* está disponível, pelo que pode ser alterado conforme as necessidades específicas de cada projecto.

Portabilidade

Esta ferramenta é inteiramente desenvolvida na plataforma $Java^{TM}$, é altamente portável e flexível. Depois de compilado, o código pode ser executado em qualquer sistema que suporte a plataforma $Java^{TM}$ [43].

Independência do orador

Com o *Sphinx-4* é possível realizar reconhecimento independente do orador, de palavras isoladas ou discurso contínuo. O reconhecimento pode ser feito sobre pequenos, médios ou

grandes vocabulários em tempo real ou diferido, com taxas de sucesso que podem chegar aos 95%.

Flexibilidade na escolha do modelo de linguagem

Esta ferramenta permite a utilização de vários tipos de modelos de linguagem, tais como, *statistical N-grams, context free grammar e finite state grammar*. Isto é possível porque a *Sphinx-4* tem um módulo, o *Graph construction*, que traduz os vários modelos num modelo interno ao descodificador [43].

Possibilidade de introduzir regras linguísticas

Não foi possível obter informação àcerca deste assunto. Contudo, tal como no HTK, deve ser possível introduzir algum tipo de regras linguísticas nos modelos.

Possibilidade de treinar o reconhecedor ao longo do tempo

Neste ponto é necessário considerar duas situações distintas. Na primeira, supõe-se que o reconhecedor está a correr numa máquina Unix (Linux). Neste caso é possível treinar o reconhecedor sempre que seja necessário. O mesmo não acontece, pelo menos com a mesma facilidade, quando o reconhecedor está instalado numa máquina Windows. Isto acontece porque o ***Sphinx-4* tem que utilizar a ferramenta SphinxTrain para treinar o reconhecedor**. Esta foi construída para correr em sistemas Unix (Linux).

Existência de modelos acústicos para o Português ou forma de os criar

À data da realização deste trabalho, não existem modelos acústicos para o Português. Contudo, e tendo em conta que a ferramenta *Sphinx-4* pode utilizar vários tipos de modelos, estes podem ser criados e treinados com outra ferramenta e posteriormente introduzidos na *Sphinx-4*. Uma ferramenta que se pode utilizar, em máquinas Unix, é a *SphinxTrain*.

Tipo de licença

O software *Sphinx* é distribuído gratuitamente e sem restrições na sua utilização e distribuição tanto para uso particular como comercial. Terão apenas que ser cumpridas as exigências que constam no ficheiro *LICENSE*, (distribuído com o software).

Informação disponível

Existe bastante informação disponível no *site* de apoio ao projecto, assim como também nas páginas pessoais das pessoas envolvidas no desenvolvimento. Contudo, a informação existente é do tipo Javadoc, isto é, gerada automaticamente pela plataforma $Java^{TM}$; o que dificulta a sua compreensão. Outro factor que também dificulta de forma significativa a consulta da documentação é o facto desta estar dispersa e pouco estruturada.

Resumo

A ferramenta *Sphinx-4* apresenta boas características. Os seus pontos fortes são a **modularidade, portabilidade e flexibilidade ao nível dos modelos que podem ser utilizados**. Contudo, a sua utilização é mais ao nível da investigação académica e principalmente em sistemas Unix. Tendo em conta que a maior parte dos computadores existentes no mercado são máquinas Windows, a utilização da *Sphinx-4* em software comercial não é imediata. Esta **utiliza aplicações construídas para serem usadas em máquinas Unix (Linux)**, por exemplo, a aplicação utilizada para criar e treinar os modelos acústicos, o *SphinxTrain*. **Um outro ponto fraco é a documentação**. Está escrita de forma pouco amigável, isto é, dificulta a leitura a utilizadores não especializados no assunto, o que também é um entrave à utilização da *Sphinx-4*.

4.5.3 Conclusão

Após analisar as ferramentas HTK e *Sphinx-4*, é necessário decidir qual das duas se vai utilizar para construir o reconhecedor.

Tendo em conta tudo o que foi dito nos pontos anteriores, o mais sensato é utilizar o HTK. Esta escolha não elimina a possibilidade de no futuro se optar por outra ferramenta. Os factores que levam a esta escolha são vários:

É uma solução única — O HTK fornece todas as aplicações necessárias para construir o reconhecedor, sem ser necessário recorrer a software externo.

Taxa de sucesso no reconhecimento — O HTK apresenta taxas de sucesso bastante boas, em muitos casos acima de 95%.

Boa documentação — O livro *HTK Book* é um excelente elemento de estudo, aborda praticamente todos os aspectos da construção de um reconhecedor.

A licença é mais restritiva do que a do *Sphinx-4*. Contudo, como já foi dito anteriormente, **não proíbe o desenvolvimento de software comercial com base no HTK**, proibindo apenas a distribuição.

4.6 Reconhecedor de fala dependente do orador, baseado em HTK

Esta secção apresenta as principais fases da construção de um reconhecedor de fala dependente do orador, utilizando o *HTK Toolkit*. No anexo B, encontra-se uma versão mais detalhada desta secção.

A metodologia que seguimos na construção do reconhecedor de fala é a apresentada no capítulo terceiro do livro ***The HTK Book***[44], a qual contém as seguintes fases, figura 4.22:

1. Configurações e preparação dos dados
2. Criação dos modelos monofones
3. Criação dos modelos trifones
4. Avaliação
5. Reconhecimento em tempo real

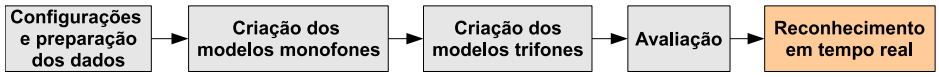


Figura 4.22: Construção do reconhecedor.

4.6.1 Configurações e preparação dos dados

Para construir um reconhecedor de fala são necessários dados acústicos, tanto para treinar como para testar o reconhecedor [44]. A recolha dos dados acústicos é a primeira tarefa a realizar na construção de um reconhecedor de fala. Para tal é necessário percorrer os seguintes passos:

1. Definir uma gramática
2. Gerar e gravar os conjuntos de frases de treino e teste
3. Construir o dicionário
4. Criar os ficheiros com a transcrição fonética
5. Extrair os *feature vectors*

Gramática

Antes de mais é necessário definir o **cenário** onde o reconhecedor vai ser utilizado. Neste caso em concreto, o cenário é uma **casa de habitação**, onde o que se pretende é executar acções sobre os **diversos dispositivos**, isto é, controlar o **ambiente envolvente**. A figura 4.23, apresenta o cenário típico.

No cenário apresentado pela figura 4.23 são vários os dispositivos que pretendemos controlar. Em concreto, podemos enumerar os seguintes: **portas, lâmpadas, tomadas, estores e o auto-clismo**. A acções que se podem executar sobre estes dispositivos são: **abrir/fechar, ligar/desligar e subir/descer**. Assim sendo, o conjunto de frases que se pretende reconhecer é o seguinte:

- Luz da Sala
- Luz da Sala dois

Figura 4.23: Cenário de utilização do reconhecedor.

- Luz da Cozinha
- Luz da Casa de Banho
- Luz do Corredor
- Luz do Quarto
- Abrir a Porta da Frente
- Abrir a Porta do Quarto
- Abrir o Estore do Quarto
- Fechar a Porta da Frente
- Fechar a Porta do Quarto
- Fechar o Estore do Quarto
- Subir o Estore
- Descer o Estore
- Ligar todas as lâmpadas

- Ligar a Tomada do Quarto
- Desligar todas as lâmpadas
- Desligar a Tomada do Quarto
- Tomada do Quarto
- Autoclismo

No conjunto de frases sugerido existem algumas que não indicam qual a acção a executar. Isto acontece porque nestes casos o que se pretende é executar a acção que **inverta o estado actual** (lâmpadas), ou então porque só é possível executar **uma única acção** e neste caso está implícita (autoclismo).

A forma de definir estas frases é através de uma **gramática**. O *HTK* disponibiliza uma linguagem para definir formalmente uma gramática. A gramática da figura 4.24, define formalmente as frases enumeradas. **Os parêntesis rectos delimitam palavras opcionais e as barras verticais separam as diversas possibilidades**.

De forma a melhorar a compreensão da gramática, esta pode ser representada graficamente. A figura 4.25 apresenta todas as possibilidades permitidas pela gramática, para construir frases iniciadas pelas palavras ”Ligar” e ”Desligar”.

A forma como a gramática é apresentada na figura 4.24 é utilizada apenas por conveniência, e é uma representação de alto nível, fácil de utilizar. O *HTK* utiliza a informação contida na gramática recorrendo a uma representação de baixo nível. De facto, é usada uma rede onde estão representadas todas as palavras bem como a forma como estas se ligam entre si. Esta notação de baixo nível chama-se ***HTK Standard Lattice Format*** **(SLF)**. O *HTK* disponibiliza uma ferramenta, o ***HParse***, que constrói a rede de palavras a partir da representação de alto nível da gramática [44].

Criação dos conjuntos de frases para treino e teste

Os conjuntos de frases para treino e teste do reconhecedor podem ser obtidos a partir da gramática. Para tal, basta utilizar a ferramenta ***HSGen*** disponibilizada pelo HTK, juntamente com um **dicionário**

```
$autoclismo = AUTOCLISMO;
$tomada = TOMADA;
$locais_com_tomada = QUARTO;
$subir_descer = SUBIR | DESCER;
$objectos_sobem_descem = ESTORE;
$dispensar = DISPENSAR;
$produtos_dispensar = PRODUTO;
$numero = UM | DOIS | TRES | QUATRO;
$ligar_desligar = LIGAR | DESLIGAR;
$luz = LUZ;
$locais_com_luz = CASA DE BANHO | COZINHA | QUARTO | CORREDOR;
$abrir_fechar = ABRIR | FECHAR;
$objectos_abrem_fecham = PORTA | ESTORE;
$locais_objectos_abrem_fecham = FRENTE | QUARTO;
(
   SENT-START
   (
      ($autoclismo) |
      ($tomada [DO] $locais_com_tomada) |
      ($subir_descer [O] $objectos_sobem_descem) |
      ($dispensar [O] $produtos_dispensar $numero) |
      ($ligar_desligar TODAS [AS] LAMPADAS) |
      ($ligar_desligar [A] TOMADA [DO] QUARTO) |
      ($luz [DA] SALA $numero) |
      ($luz [DA | DO] $locais_com_luz) |
      ($abrir_fechar DISPENSADOR) |
      ($abrir_fechar [A | O] $objectos_abrem_fecham [DA | DO] $locais_objectos_abrem_fecham)
   )
   SENT-STOP
)
```

Figura 4.24: Gramática.

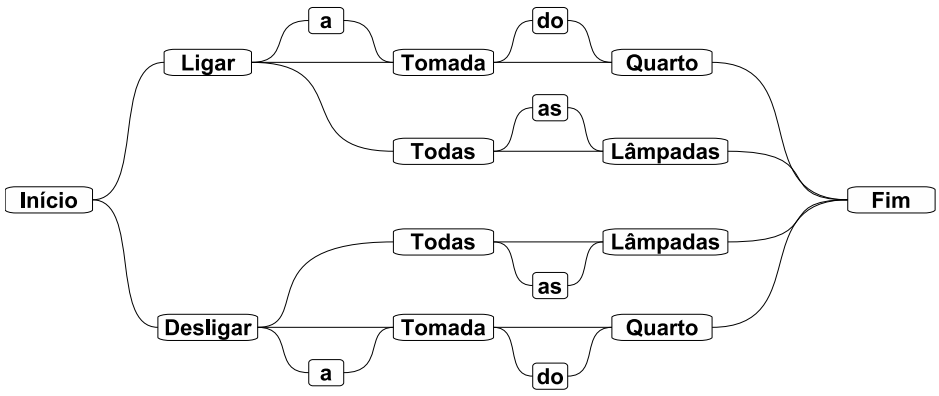


Figura 4.25: Frases iniciadas por “Ligar”e “Desligar”.

genérico para o Português. O *HSGen* gera um conjunto de N frases, aleatoriamente, usando a informação presente na gramática e no dicionário

Dicionário

O primeiro passo na construção de um dicionário é a identificação de todas as palavras presentes nas frases que se pretende reconhecer. O dicionário tem que conter a **representação fonética** de todas as palavras que constituem estas frases. A melhor forma de obter esta lista de palavras é extraí-la automaticamente do conjunto de frases de treino.

Gravação das frases para treino e teste

Neste ponto da construção do reconhecedor é necessário gravar o material de áudio correspondente às frases de treino e de teste. A gravação destas frases é feita recorrendo à ferramenta ***HSLab***, fornecida pelo *HTK*.

Criação dos ficheiros com a transcrição fonética

Para que os dados que acabamos de obter sejam úteis é necessário gerar as respectivas transcrições fonéticas. Para o efeito, vamos utilizar as ferramentas disponibilizadas pelo *HTK*. O primeiro passo é criar um ***Master Label File*** **(MLF)** com as transcrições ao nível da palavra para cada um dos ficheiros de áudio. Para criar este ficheiro de uma forma automática podemos utilizar o ***script TrainPrompts2mlf.pl*** disponibilizado juntamente com o *HTK*.

Agora que já possuímos as transcrições ao nível da palavra podemos avançar para a **transcrição fonética**. O ficheiro MLF contendo as transcrições fonéticas pode ser gerado de uma forma automática pela ferramenta ***HLEd***. O *HLEd* utiliza a informação existente no dicionário, em conjunto com as transcrições ao nível da palavra, para gerar as transcrições fonética.

Extracção dos *feature vectors*

Esta é a última tarefa a realizar no que diz respeito à preparação dos dados. Consiste na extracção de vectores com as características mais relevantes do sinal, tendo em conta a tarefa que se pretende realizar. Neste caso concreto pretende-se extrair as características mais relevantes para reconhecimento de fala. Na literatura de língua Inglesa estes vectores têm o nome de ***feature vectors***.

A extracção destes vectores pode ser feita de uma forma automática utilizando a ferramenta ***HCopy***, fornecida pelo *HTK*.

4.6.2 Criação dos modelos monofones

Nesta fase da criação do nosso reconhecedor o que se pretende é obter um **conjunto de modelos monofones bem treinados**. Para tal, é necessário executar as seguintes tarefas: **inicialização dos modelos, ajuste dos modelo de silêncio, introdução do modelo para pausas curtas e realinhamento dos dados**. Em cada uma destas tarefas é necessário refinar os modelos. Isto é feito através da re-estimação dos parâmetros dos mesmos através do método de *Baum-Welch*, utilizando a ferramenta *HERest*.

Inicialização dos modelos monofones

O primeiro passo no sentido de treinar um conjunto de HMM, é criar o protótipo dos modelos. Neste primeiro passo o mais importante é definir o modelo e não os seus parâmetros. Nos sistemas que têm por unidade base o fonema é usual utilizar uma **topologia *left-right* com 3 estados** [44]. A figura 4.26 ilustra a topologia referida, onde os vectores de médias e variâncias têm comprimento igual a 39, isto é 13 coeficientes MFCC mais 13 coeficientes delta mais 13 coeficientes de aceleração.

Depois de definir o protótipo para os HMM é necessário inicializá-lo. A inicialização do protótipo consiste em substituir o valor zero presente nos vectores de médias e o valor um dos vectores de variâncias pelos valores globais média e variância, respectivamente. Isto pode ser feito pela ferramenta ***HCompV***.

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
   <NumStates> 5
   <State> 2
   <Mean> 39
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
   <Variance> 39
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
   <State> 3
   <Mean> 39
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
   <Variance> 39
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
   <State> 4
   <Mean> 39
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
   <Variance> 39
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Figura 4.26: Protótipo para os HMMs.

Agora que já temos um ponto de partida para a criação dos modelos, **é necessário proceder à re-estimação** dos mesmos até que sejam suficientemente robustos. A ferramenta ***HERest*** permite efectuar esta tarefa.

Ajuste do modelo de silêncio e introdução de pausas curtas

Nos passos anteriores foram gerados modelos com três estados para cada um dos fonemas em uso, bem como para o **modelo de silêncio *sil***. O modelo de silêncio existente não permite a ocorrência de estados de silêncio consecutivos, pelo que é pouco robusto. Para solucionar este problema vamos alterar o modelo de silêncio de forma que seja possível existir transições entre os estados dois e quatro em ambos os sentidos.

Vamos introduzir também o **modelo para pequenas pausas *sp***. O modelo *sp* tem apenas **um estado**, que corresponde ao estado central do modelo de silêncio.

Realinhamento dos dados

Os modelos existentes foram estimados prevendo a existência de pequenas pausas entre as palavras que constituem as frases que se pretende reconhecer, o que é uma boa aproximação à realidade. Contudo, as transcrições existentes foram geradas partindo do principio que as palavras estavam encostadas umas às outras, ou seja, não foi prevista a existência de pequenas pausas entre as palavras que constituem as frases. Para solucionar este problema temos que introduzir as pequenas pausas existentes nos modelos acústicos na transcrição fonética.

4.6.3 Criação dos modelos trifones

Neste passo o que se pretende é construir modelos dependentes do contexto, neste caso trifones. Os modelos dependentes do contexto contêm informação àcerca dos **sons vizinhos**, pelo que apresentam melhores resultados no reconhecimento. **Os modelos trifones são criados a partir dos modelos monofones existentes**.

Criação dos modelos trifones a partir dos monofones

Antes de mais é necessário criar **novas transcrições fonéticas**, substituindo os monofones pelos trifones correspondentes. Isto pode ser feito com a ferramenta ***HLEd***. É também necessário criar novos modelos tendo em conta as transcrições com trifones. Estes podem ser gerados automaticamente a partir dos que já existem.

4.6.4 Avaliação do reconhecedor

A avaliação dos modelos é fundamental para verificar se podemos parar o processo de treino ou, pelo contrário, se devemos continuar. A avaliação irá ser feita tanto aos modelos monofones como aos trifones. A ferramenta disponibilizada pelo *HTK* para avaliar os modelos criados é o ***HResults***. O *HResults* compara as transcrições obtidas no processo de reconhecimento, com as correctas (manuais) dando informação acerca da percentagem de frases correctamente reconhecidas, WER, eliminações, substituições e inserções.

4.6.5 Reconhecimento em tempo real

O reconhecimento em tempo real faz-se recorrendo à ferramenta *HVite*, disponibilizada com o *HTK*. O *HVite* utiliza o algoritmo de *Viterbi* para determinar qual o conjunto de HMMs, que tem maior probabilidade de ter gerado os dados acústicos em análise.

4.7 Reconhecedor de fala independente do orador

O **reconhecedor de fala independente do orador** foi desenvolvido a partir dos exemplos fornecidos com o ***Microsoft Speech Recognition Sample Engine for Portuguese (Portugal)***, disponibilizado pelo ***Microsoft Language Development Center* (MLDC)** [15]. A tecnologia utilizada é a ***Microsoft Speech API 5.3* (SAPI)**. **O *Microsoft Speech Recognition Sample Engine for Portuguese (Portugal)* já contém os modelos acústicos para o Português**, pelo que a construção do reconhecedor fica bastante simplificada.

Ao mais alto nível, a SAPI dispõe de classes e interfaces para interagir com os diferentes *speech engines* que disponibiliza. A baixo nível é responsável pelo controlo e gestão dos *speech engines*, sejam eles para reconhecimento ou síntese de fala. Como se pode ver a partir da figura 4.27, **as aplicações não têm acesso directo aos *speech engines***, fazem-no através de um ***runtime componet*** o SAPI Runtime [45].

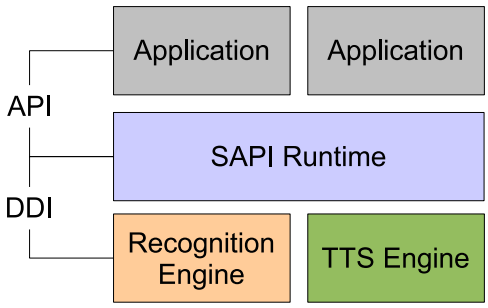


Figura 4.27: Arquitectura da Microsoft Speech API 5.3. Múltiplas aplicações podem partilhar os *speech engines* disponíveis através da SAPI *Runtime*.

Para desenvolver aplicações com reconhecimento de fala utilizando esta tecnologia, é necessário definir três componentes base: **o *recognition context*, o reconhecedor e a gramática**.

Contextos de reconhecimento

As principais vias de comunicação entre uma aplicação com reconhecimento de fala e a SAPI, são os contextos de reconhecimento (*recognition contexts*). É através deles que as aplicações controlam os reconhecedores de fala e recebem os eventos por eles gerados. Neste trabalho apresentamos apenas os aspectos relacionados com reconhecimento de fala. As questões relacionadas com a síntese não serão abordadas. Em aplicações de reconhecimento de fala os *recognition contexts* podem ser de dois tipos, ***shared contexts*** ou ***in process (InProc) contexts*** [45].

A utilização de um *shared context* permite que os recursos disponíveis, placa de som, reconhecedor e gramáticas, sejam partilhados por várias aplicações ou *recognition contexts*. Neste caso, sempre que o reconhecedor produz um resultado, a SAPI envia-o para o *recognition context* cuja gramática melhor resultado oferece na taxa de reconhecimento. A declaração de *shared contexts* é feita recorrendo à classe ***SpSharedRecoContext*** [45].

O *InProc context* restringe a utilização dos recursos existentes a um único *recognition context* ou aplicação. Isto é, a placa de som, o reconhecedor e a gramática em uso por um *InProc context* não podem ser utilizados por mais nenhuma aplicação ou *recognition context*. **Aplicações com requisitos elevados ao nível do tempo de resposta do reconhecedor e exactidão no reconhecimento devem utilizar *InProc contexts***. Para declarar *InProc contexts* recorre-se à classe ***SpInProcRecoContext*** [45].

Tendo em conta os requisitos da interface que pretendemos desenvolver, nomeadamente, elevada taxa de sucesso no reconhecimento e rapidez na resposta, **o *InProc context* apresenta-se como sendo o mais adequado**.

Reconhecedores de fala

A SAPI permite criar dois tipos de reconhecedores de fala, ***SpInprocRecognizer*** e ***SpSharedRecognizer***. A utilização de um ou outro depende do *recognition context* que se pretende utilizar. Neste trabalho vai-se utilizar um reconhecedor do tipo *SpInprocRecognizer*, isto porque, como já vimos no ponto anterior, pretendemos fazer reconhecimento num *InProc context*. Os *recognition contexts* podem ser criados pelos reconhecedores, para tal basta executar o seguinte código [46]:

```
(1) SpInprocRecognizer Recognizer = new SpInprocRecognizer();
...
(2) ISpeechRecoContext recoContext = Recognizer.CreateRecoContext();
(3) RC = (SpInProcRecoContext)recoContext;
```

O código anterior exemplifica a declaração e criação de um *SpInprocRecognizer* (linha 1). De seguida exemplifica-se a declaração e criação de um *SpInProcRecoContext* (RC) utilizando o reconhecedor previamente criado (linhas 2 e 3).

Neste momento já vimos como se criam reconhecedores e *recognition contexts*. Falta apenas um componente para podermos fazer reconhecimento de fala, a gramática. **A gramática é criada a partir do *recognition context***. Para tal, pode-se utilizar o excerto de código que se segue [46]:

```
(1) internal ISpeechRecoGrammar grammar;
...
(2) grammar = RC.CreateGrammar(1);
(3) grammar.DictationLoad("", SpeechLib.SpeechLoadOption.SLOStatic);
(4) grammar.DictationSetState(SpeechLib.SpeechRuleState.SGDSInactive);
```

Este código exemplifica a declaração e criação de uma gramática (linhas 1 e 2). De seguida, define-se se o vocabulário existente na gramática pode ser alterado ou não, neste caso não se permitindo alterações (linha 3). Por fim, define-se se a gramática acabada de criar pode ser utilizada, de imediato, no reconhecimento ou não (linha 4).

Gramáticas

As palavras ou conjuntos de palavras que pretendemos reconhecer têm que estar definidos numa gramática. Na SAPI 5 as gramáticas são implementadas como *context-free grammars* (CFGs). De uma forma muito simples, podemos dizer que uma CFG define quais as palavras ou conjuntos de palavras que um determinado reconhecedor de fala pode reconhecer.

Na SAPI 5 as CFGs são definidas utilizando *Extensible Markup Language* (XML). A figura 4.28, apresenta a estrutura base de uma CFG. A figura 4.29, apresenta um excerto da gramática utilizada pelo reconhecedor de fala.

```
<GRAMMAR LANGID="">
   <DEFINE>
      <ID NAME="" VAL=""/>
   </DEFINE>
   <RULE NAME="" TOPLEVEL="">
      <L PROPNAME="" PROPID="">
         <P> </P>
         <P> </P>
           ...
      </L>
   </RULE>
</GRAMMAR>
```

Figura 4.28: Estrutura da gramática utilizada pela SAPI 5.3.

```
<GRAMMAR LANGID="100">
  <DEFINE>
     <ID NAME="blive" VAL="1"/>
  </DEFINE>
  <RULE NAME="blive" TOPLEVEL="ACTIVE">
     <L PROPNAME="blive" PROPID="blive">
        <P>LUZ DA SALA</P>
        <P>LUZ SALA</P>
        <P>LUZ DA SALA DOIS</P>
        <P>LUZ SALA DOIS</P>
        <P>LUZ DA COZINHA</P>
        <P>LUZ COZINHA</P>
        <P>LUZ DA CASA DE BANHO</P>
                  ...
```

Figura 4.29: Gramática utilizada pelo reconhecedor independente do orador.

A SAPI dispõe de ferramentas para carregar as gramáticas a partir dos seus ficheiros, e também de compiladores. O compilador de CFGs transforma o ficheiro XML em formato binário. Depois de compiladas as gramáticas podem ser utilizadas pelos reconhecedores.

4.8 Comentários finais

Como vimos ao longo deste capítulo, na implementação desta interface foi necessário desenvolver várias aplicações (o reconhecedor, a base de dados e a aplicação de interface), cada uma delas com uma tarefa específica.

A aplicação de interface é responsável por interagir com o reconhecedor, com a base de dados e com o sistema domótico B-LIVE. Para que esta interacção seja possível foi necessário desenvolver vário software. Durante o desenvolvimento deste software foi necessário ultrapassar algumas dificuldades, tais como: como lançar, terminar e receber dados de uma aplicação a partir do Java. **Foi necessário desenvolver software para aceder ao hardware (porta série) e à base de dados**. As **comunicações com o B-LIVE** foram implementadas de acordo com as suas especificações, as quais foram apresentadas apenas naquilo que é essencial para perceber o funcionamento da interface.

No desenvolvimento da base de dados foi necessário definir com algum cuidado as **relações entre as tabelas**, assim como os campos que constituem cada uma delas. A utilização de ***Stored Queries*** cria alguma independência em relação ao motor de base de dados que se está a utilizar.

O reconhecedor dependente do orador foi desenvolvido com a ferramenta *HTK*. As razões que levaram à escolha desta ferramenta foram apresentadas e discutidas com algum pormenor. Esta ferramenta é responsável pela transformação dos sinais acústicos de fala em texto. **As principais dificuldades encontradas durante o seu desenvolvimento foram a criação do dicionário e a recolha de dados para teste e treino**. Por fim, apresentamos os passos necessários para construir um reconhecedor independente do orador, utilizando o *Microsoft Speech Recognition Sample Engine for Portuguese (Portugal)* distribuído pelo MLDC.

Capítulo 5

Resultados

Os resultados da avaliação feita à interface desenvolvida podem ser divididos em dois níveis: **resultados técnicos e resultados de usabilidade**.

Os resultados técnicos estão relacionados com o desempenho da interface, mais concretamente dos reconhecedores de fala. Estes resultados são apresentados e analisados em pormenor nas secções 5.1 e 5.2.

A usabilidade foi avaliada por utilizadores com limitações funcionais em tratamento no CMRRC-Rovisco Pais. Esta avaliação fez-se através do preenchimento de um questionário, depois de terem utilizado a interface. Os resultados obtidos são apresentados e discutidos na secção 5.3.

5.1 Avaliação dos resultados obtidos com o reconhecedor dependente do orador

A interface pode ser integrada com dois tipos de reconhecedores, dependentes ou independentes do orador. Nesta secção apresentamos os resultados obtidos com a interface integrada com o reconhecedor **dependente do orador**[1] desenvolvido com o *HTK*. Os testes foram efectuados em **ambiente controlado (laboratório)**, utilizando 100 frases geradas automaticamente pelo **HSGen**.

[1] Por conveniência o utilizador desta versão é o autor deste trabalho.

Estas frases não usadas no treino do reconhecedor.

5.1.1 Reconhecimento com modelos monofones

A figura 5.1, apresenta o relatório da avaliação feita aos modelos monofones. Analisando-o podemos verificar que os resultados são bastante bons. A percentagem de frases correctamente reconhecidas ($SENT : \%Correct$) é de 97.00%, e quanto à percentagem de palavras correctamente reconhecidas ($WORD : \%Corr$) obtivemos um resultado de 100.00%. Estes resultados têm uma precisão na taxa de reconhecimento de palavras de 99.00%. A diferença entre $SENT : \%Correct$ e $WORD : \%Corr$ deve-se a 3 inserções ($I = 3$). Estes resultados foram obtidos com o comando apresentado em B.4.1.

```
====================== HTK Results Analysis =======================
  Date: Mon Oct 22 11:36:27 2007
  Ref : TestWords.mlf
  Rec : recout_mono.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=97.00 [H=97, S=3, N=100]
WORD: %Corr=100.00, Acc=99.00 [H=301, D=0, S=0, I=3, N=301]
===================================================================
```

Figura 5.1: Relatório da avaliação feita aos modelos monofones.

5.1.2 Reconhecimento com modelos trifones

Os resultados da avaliação feita aos modelos trifones são apresentados na figura 5.2. Também estes são resultados bastante positivos, já que obtivemos 97.00% na percentagem de frases correctamente reconhecidas (valor igual ao obtido para os modelos monofones). A percentagem de palavras correctamente reconhecidas é de 99.67% e a precisão com que foram reconhecidas é de 99.00%. A diferença entre os valores de $SENT : \%Correct$ e $WORD : \%Corr$ deve-se a 1 eliminação e 2 inserções ($D = 1, I = 2$). Estes resultados foram obtidos com o comando apresentado em B.4.2.

```
====================== HTK Results Analysis =======================
  Date: Mon Oct 22 11:38:53 2007
  Ref : TestWords.mlf
  Rec : recout_tri.mlf
------------------------ Overall Results --------------------------
SENT: %Correct=97.00 [H=97, S=3, N=100]
WORD: %Corr=99.67, Acc=99.00 [H=300, D=1, S=0, I=2, N=301]
===================================================================
```

Figura 5.2: Relatório da avaliação feita aos modelos trifones.

5.1.3 Reconhecimento em tempo real

O procedimento utilizado para testar o reconhecimento em tempo real foi o seguinte: **repetiram-se aleatoriamente as 28 frases que o reconhecedor deve reconhecer (tabela C.1 do anexo C) 15 vezes cada uma, registando-se se o reconhecimento foi efectuado com sucesso ou não**. O sucesso significa que toda a frase foi reconhecida correctamente.

O gráfico da figura 5.3 apresenta os resultados obtidos. As colunas a **verde** indicam o número de vezes que cada uma das frases foi reconhecida correctamente, a **laranja** representa-se o número de vezes em que o reconhecimento foi incorrecto.

Como se pode ver a partir do gráfico da figura 5.3, o desempenho global do reconhecedor é bastante bom. **A taxa de sucesso de frases correctamente reconhecidas é de** 92.80%[2]. Analisando os resultados em mais pormenor verifica-se que ocorreram erros apenas no reconhecimento de três frases: ***"Luz da Sala um"***, ***"Dispensar o produto um"*** **e** ***"Dispensar o produto três"***. A taxa de sucesso no reconhecimento destas frases é bastante baixa, 60.00%, 46.67% e 6.67%, respectivamente.

Após uma análise mais atenta às frases com erros no reconhecimento, verifica-se que estas contêm palavras que não se repetem em mais nenhuma frase. É o caso da palavra ***"um"*** (nas frases: *"Luz da Sala um"* e *"Dispensar o produto um"*) e da palavra ***"três"*** (na frase *"Dispensar o produto três"*). Isto evidencia um possível problema com os dados utilizados para treinar os

[2]Total de frases é 390, total de frases correctamente reconhecidas é 362. Logo: $(\frac{362}{390} * 100 = 92.80\%)$.

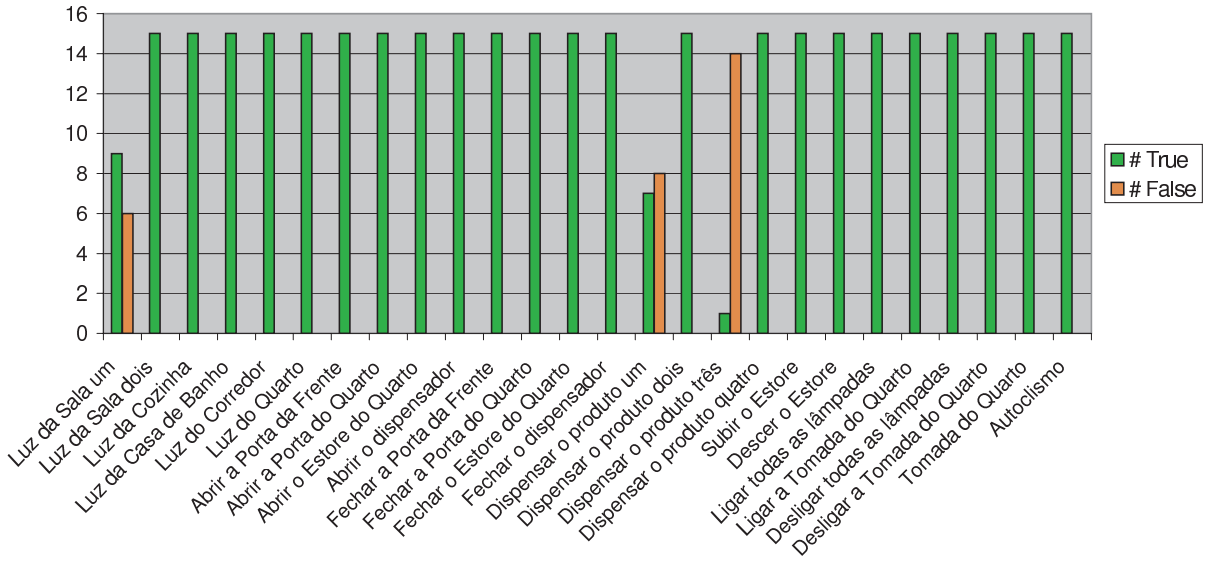


Figura 5.3: Resultados obtidos com o reconhecedor dependente do orador no reconhecimento em tempo real.

modelos, ou seja, **podem existir poucos dados acústicos representativos das palavras *"um"* e *"três"***. A possível escassez de dados e consequentemente modelos pouco robustos podem explicar o fraco desempenho no reconhecimento destas frases.

5.2 Avaliação dos resultados obtidos com o reconhecedor independente do orador

A avaliação do desempenho do reconhecedor independente do orador fez-se recorrendo a um grupo de 10 pessoas[3]. Os dados relativos à naturalidade, sexo e idade destas pessoas estão representados na figura 5.4. Quanto à naturalidade, os utilizadores são provenientes de quatro distritos (Aveiro, Coimbra, Guarda e Porto). Relativamente ao sexo os utilizadores são maioritariamente homens, havendo apenas duas mulheres. As idades variam entre os 22 e os 38 anos.

O procedimento de teste utilizado foi o seguinte, **cada uma das pessoas do grupo de teste repetiu uma única vez, de forma aleatória as frases da tabela C.1 do anexo C**. Para cada uma das frases registou-se se o resultado do reconhecimento foi correcto ou não. O gráfico da figura 5.5,

[3]O autor deste trabalho não está incluído neste grupo.

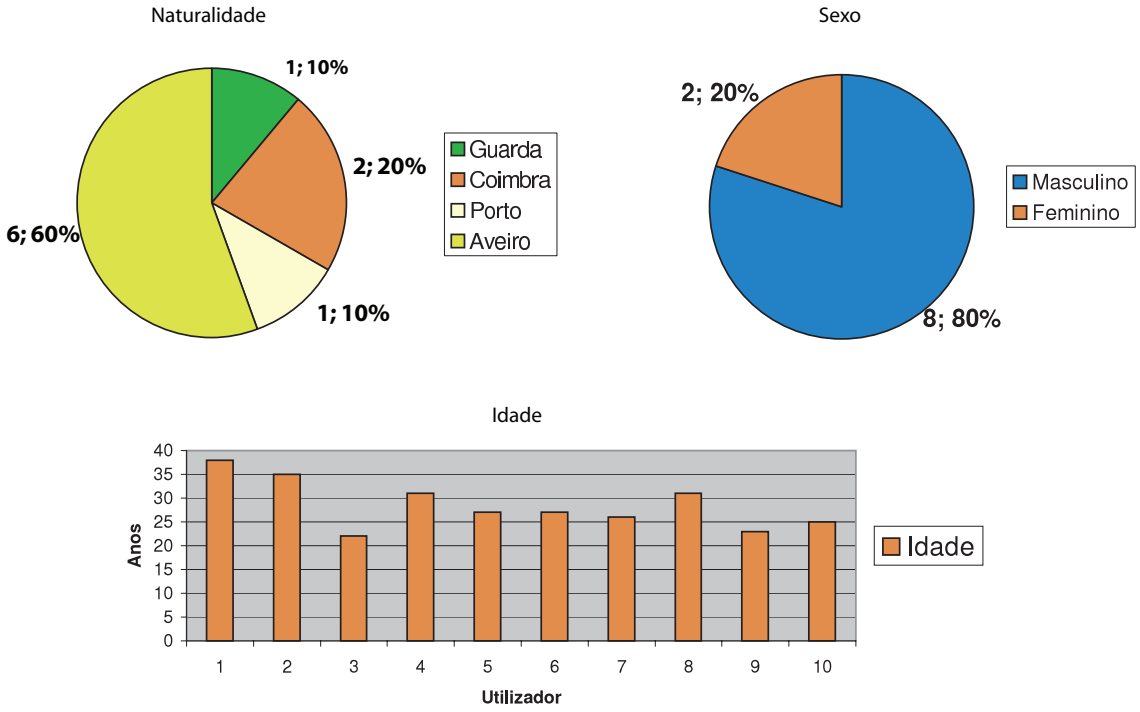


Figura 5.4: Dados sobre a naturalidade, sexo e idade dos utilizadores que avaliaram o desempenho do reconhecedor independente do orador.

apresenta os resultados obtidos. Os testes realizaram-se em ambiente laboratorial, onde estavam 7 pessoas a trabalhar (sem alterarem as suas rotinas diárias).

Analisando os resultados verifica-se que a taxa de sucesso de frases correctamente reconhecidas é de 91.54%. Este valor é bastante próximo do que tínhamos obtido com o reconhecedor dependente do orador. Analisando o gráfico da figura 5.5, verifica-se que a frase *"Dispensar o produto três"* continua com uma taxa de erro muito elevada, 70%. Os restantes erros aparecem de uma forma distribuída pelas restantes frases.

5.3 Avaliação dos resultados obtidos em utilização real no CMRRC-Rovisco Pais

A avaliação da interface em utilização real fez-se nas instalações do Centro de Medicina de Reabilitação da Região Centro - Rovisco Pais (CMRRC - Rovisco Pais), local onde está instalado

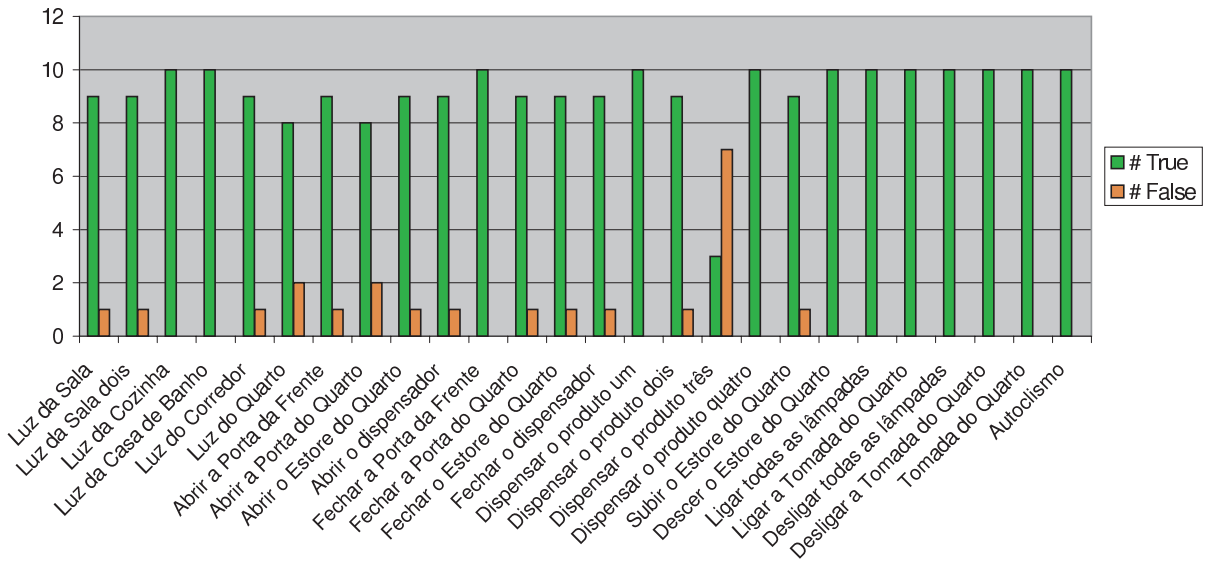


Figura 5.5: Resultados obtidos no reconhecimento em tempo real, com o reconhecedor independente do orador.

um demonstrador real do sistema domótico B-LIVE. O acesso às instalações realizou-se no âmbito de uma parceria entre o CMRRC - Rovisco Pais, a Universidade de Aveiro e a Micro I/O.

Os ensaios à interface realizaram-se com dois utilizadores [4], um do sexo masculino com 29 anos de idade e com lesão medular ao nível das vértebras **C5**/**C6** (**utilizador M**) e outro do sexo feminino com 40 anos de idade e cuja lesão medular é ao nível das vértebras **C4**/**C5** (**utilizador F**). O utilizador F apresenta algumas especificidades ao nível da voz, nomeadamente: sotaque Brasileiro, baixa intensidade (dificuldade em falar alto) e tom grave (tendo em conta o seu sexo). O acompanhamento médico destas pessoas foi assegurado pela Drª Arminda Lopes, médica fisiatra do CMRRC-Rovisco Pais.

Os ensaios foram realizados tendo em conta os cenários de utilização descritos em C.1 e C.2. Ambos os cenários pretendem retratar situações que ocorrem, com bastante frequência, no dia-a-dia das pessoas. O cenário C.1, retrata uma situação em que o utilizador necessita de se deslocar no interior da habitação entre as diversas divisões. No cenário C.2, pretende-se simular uma ida ao exterior: o utilizador está no interior da habitação e sente necessidade de sair. A descrição

[4] A avaliação da interface em larga escala é inviável devido à logística necessária para deslocar e acompanhar as pessoas envolvidas. Neste caso em concreto, foi necessário providenciar ambulâncias de transporte, acompanhamento médico e familiar, refeições, etc.

pormenorizada dos cenários e do procedimento utilizado nos ensaios pode ser consultada no anexo C.

A atribuição dos cenários aos utilizadores não seguiu nenhum critério, foi aleatória. Ao utilizador M foi atribuído o **cenário C.1** e ao utilizador F o **cenário C.2**.

O utilizador M realizou a maioria das tarefas propostas no cenário C.1 sem problemas. À excepção de "*Autoclismo*", todos os comandos foram reconhecidos com sucesso, sem necessidade de repetição. O utilizador F teve alguns problemas em realizar as tarefas propostas no cenário C.2. Os comandos "*Desligar a tomada do quarto*" e "*Desligar a luz do quarto*" não foram reconhecidos mesmo após várias repetições. Ao contrário, os comandos "*Abrir a porta da frente*" e "*Fechar a porta da frente*" foram reconhecidos sem problemas. **As dificuldades encontradas com o utilizador F podem ser explicadas, em parte, pelo facto deste ter sotaque brasileiro**.

No final dos ensaios foi realizado um pequeno questionário, apresentado no anexo C. Tendo em conta as respostas dadas pelos utilizadores M e F, pode-se concluir que a interface com reconhecimento de fala correspondeu às expectativas dos mesmos. Nas suas respostas salientaram a simplicidade e facilidade de utilização da interface, características que constam dos objectivos do trabalho. Abordaram também aspectos relacionados com a maior autonomia e mobilidade que a interface oferece, principalmente por permitir que as mão fiquem livres para efectuar outras tarefas, por exemplo, manipular a cadeira eléctrica. Fizeram também algumas sugestões, tais como, adicionar funcionalidades que permitam controlar electrodomésticos, o vídeo porteiro e o elevador.

Não podemos deixar de referir que características como simplicidade, facilidade de utilização, maior autonomia e mobilidade, são diferenciadoras das interfaces de fala. Pelo que, é bastante gratificante que tenham sido reconhecidas e referidas pelos utilizadores. **As respostas ao questionário, apresentadas em C.4, foram transcritas por uma terceira pessoa**, não tendo o autor deste trabalho qualquer influência no processo, para além do desenvolvimento do próprio inquérito.

Tendo em conta os ensaios realizados e as especificidades do utilizador F, pode-se dizer que o funcionamento do reconhecedor de fala em ambiente real é bastante aceitável. **As dificuldades encontradas pelo utilizador F podem, em princípio, ser minoradas procedendo à adaptação dos modelos acústicos**.

5.4 Comentários finais

Neste capítulo apresentamos e discutimos os resultados obtidos nas avaliações feitas à interface desenvolvida.

Na avaliação feita ao reconhecedor de fala dependente do orador verificou-se que o seu funcionamento é bastante bom. Contudo, pode ser melhorado, para tal, basta ter o cuidado de recolher dados para treino dos modelos que representem igualmente todos os sons.

O reconhecedor de fala independente do orador foi avaliado em dois ambientes distintos, em laboratório e em utilização real. Os resultados obtidos em laboratório foram bastante bons, muito semelhantes aos do reconhecedor dependente do orador. Em utilização real o reconhecedor correspondeu às expectativas. No entanto, em alguns casos específicos pode ser necessário fazer adaptação dos modelos acústicos.

Capítulo 6

Conclusões

O desenvolvimento da interface com reconhecimento de fala que propomos neste trabalho passou por diversas fases, em cada uma delas foi necessário tomar decisões e resolver os problemas que iam surgindo. **Neste capítulo fazemos um resumo do trabalho efectuado**, no qual pretendemos evidenciar as principais tarefas realizadas e quais as decisões que foram tomadas.

No capítulo anterior foram analisadas as avaliações técnicas e de usabilidade da interface desenvolvida. Neste capítulo iremos analisar os resultados obtidos de um ponto de vista mais abrangente, **avaliando o impacto que a interface pode ter na vida dos seus utilizadores**.

Por fim, são feitas algumas sugestões no sentido de **melhorar a interface**, quer para a tornar **mais robusta** quer para introduzir **novas capacidades e funcionalidades**.

6.1 Resumo do Trabalho

No resumo aqui apresentado pretendemos abordar as principais tarefas da realização deste trabalho. As tarefas serão apresentadas pela ordem que foram realizadas.

6.1.1 Tecnologias disponíveis no mercado

O caminho a percorrer durante o desenvolvimento de um projecto, depende das tecnologias disponíveis para a sua realização. É necessário conhecê-las para se poder optar por um caminho que permita obter resultados. No caso concreto deste trabalho o estudo das tecnologias existentes foi feito a dois níveis: **tecnologias baseadas em PC e tecnologias baseadas em microprocessador**.

Após estudar algumas soluções baseadas em microprocessador tais como: o *VR Stamp* da *Sensory* [47] e o *Kit CK5000* da *Parrot* [48], **verificamos que estas soluções não são adequadas para o projecto que pretendemos desenvolver**. As principais limitações são o limitado número de frases que conseguem reconhecer e a dificuldade em construir os modelos acústicos para o Português.

Quanto às tecnologias baseadas em PC, estas podem ser separadas em dois grupos: comerciais e de utilização livre. Para limitar os custos do projecto optamos por descartar à partida as tecnologias comerciais, pelo que apenas foram estudadas soluções de utilização livre. Para a construção do reconhecedor de fala dependente do orador, seleccionamos duas para estudar em mais pormenor, o *HTK* [40] e o *Sphinx 4* [42]. Quanto ao reconhecedor independente do orador, a ferramenta escolhida foi a fornecida pela Microsoft.

6.1.2 Escolha da ferramenta de reconhecimento a utilizar

O estudo sobre a tecnologia disponível para construir o reconhecedor de fala dependente do orador, resultou na selecção de duas ferramentas, o *HTK* e o *Sphinx 4*. O passo seguinte foi verificar qual destas ferramentas era a mais indicada para construir o reconhecedor, tendo em conta as necessidades do projecto.

Estas ferramentas foram avaliadas tendo em conta os seguintes pontos: **acesso ao código fonte, portabilidade, independência do orador, flexibilidade na escolha do modelo da linguagem, possibilidade de introduzir regras linguísticas, existência de modelos acústicos para o Português ou forma de os criar, possibilidade de treinar o reconhecedor ao longo do tempo, tipo de licença e informação disponível**. Após analisar estes pontos em ambas as ferramentas, **optamos por escolher o *HTK***.

Para construir o **reconhecedor independente do orador** utilizou-se a ferramenta disponibilizada pelo *Microsoft Language Development Center* (MLDC), o *Microsoft Speech Recognition Sample Engine for Portuguese (Portugal)*. Esta ferramenta ficou disponível para utilização experimental já no decorrer deste trabalho.

6.1.3 Bases teóricas sobre reconhecimento de fala

O processo de escolha da ferramenta de reconhecimento a utilizar para construir o reconhecedor está concluído. **Antes de avançar para a criação do reconhecedor foi feito um estudo teórico sobre reconhecimento de fala**. Neste estudo abordaram-se os seguintes assuntos: **definição do problema de reconhecimento, componentes de um reconhecedor típico, processamento do sinal acústico de fala, modelos da linguagem, modelos acústicos, descodificador e avaliação**.

Uma vez que a maioria dos sistemas de reconhecimento actuais são baseados em **modelos de Markov não observáveis** (o *HTK* é um deles), foi feita uma pequena introdução a este assunto a quando do estudo dos modelos acústicos.

6.1.4 Limitações das pessoas com tetra e paraplegia em produzir sons de fala

Os utilizadores alvo da interface aqui desenvolvida são pessoas com limitações funcionais, tetra e paraplégicos. Assim sendo, foi necessário compreender se estas lesões afectam a capacidade destas pessoas produzirem sons de fala. **Foram analisados os processos de produção e percepção dos sons de fala**, bem como a coordenação e controlo dos sistemas produtor e auditivo. Do estudo efectuado concluiu-se que as pessoas com limitações funcionais podem presentar algumas dificuldades em produzir sons de fala, as mais comuns são: **cansaço, rouquidão, baixa amplitude dos sons produzidos e dificuldade em colocar a voz**.

6.1.5 Desenvolvimento da aplicação de interface

A aplicação de interface é ponto chave de todo o sistema, sendo a responsável por interagir com o reconhecedor de fala e com o sistema domótico B-LIVE. O primeiro passo no desen-

volvimento desta interface foi definir a sua **arquitectura**. De seguida, foi necessário desenvolver cada um dos módulos que a constituem. A informação necessária ao funcionamento da interface foi estruturada e armazenada numa **base de dados relacional**.

A arquitectura utilizada permitiu dividir o código em camadas: ***Interface Layer***, ***Business Layer***, ***Hardware Layer***, ***Data Base Layer***, ***External Connection Layer*** e ***Recognizer Layer***. Cada uma destas camadas é responsável pela realização de uma tarefa bem definida.

6.1.6 Desenvolvimento dos reconhecedores de fala

Foram construídas duas versões do reconhecedor de fala, uma dependente e outra independente do orador. A versão dependente do orador foi desenvolvida utilizando o HTK, seguindo o tutorial existente em [44]. O reconhecedor independente do orador foi desenvolvido a partir dos exemplos fornecidos com o *Microsoft Speech Recognition Sample Engine for Portuguese (Portugal)*, disponibilizado pelo *Microsoft Language Development Center* (MLDC).

A tarefa mais morosa e critica no desenvolvimento do reconhecedor dependente do orador foi a recolha de dados. De forma a minimizar os custos desta tarefa, a recolha dos dados necessários para treinar o reconhecedor foi efectuada pelo autor deste trabalho.

Inicialmente estava previsto desenvolver o reconhecedor de fala independente do orador com o HTK. No entanto, durante o desenvolvimento deste trabalho a Microsoft disponibilizou o *Microsoft Speech Recognition Sample Engine for Portuguese (Portugal)*, o que levou a uma alteração dos planos iniciais. Ao utilizar esta ferramenta evitamos a tarefa de recolha de dados para treino do reconhecedor, uma vez que os **modelos acústicos são fornecidos**. Contudo, nos casos em o reconhecedor tem mau desempenho, a **ausência de ferramentas para treinar ou adaptar os modelos acústicos fornecidos pode inviabilizar a utilização desta ferramenta**.

6.1.7 Avaliação da interface

A avaliação da interface foi feita a dois níveis: **avaliação do reconhecedor e avaliação da usabilidade da interface**.

A avaliação ao reconhecedor foi feita, numa primeira fase, com dados pré gravados, recorrendo às ferramentas disponibilizadas pelo *HTK* e, posteriormente, com reconhecimento em tempo real, tanto em laboratório como em utilização num cenário real. **A versão do reconhecedor testada em cenário real foi a independente do orador**. Desta forma avaliamos o comportamento do reconhecedor quando utilizado por pessoas que não participaram no processo de recolha de dados para treino.

A avaliação de usabilidade foi feita com a colaboração de pessoas com limitações funcionais, em utilização real no Medicina de Reabilitação da Região Centro - Rovisco Pais. A avaliação foi feita através do preenchimento de um relatório após utilização da interface.

6.2 Principais Resultados

O objectivo deste trabalho foi desenvolver uma interface com reconhecimento de fala para pessoas com limitações funcionais (tetra e paraplégicos), **objectivo este que foi alcançado**. A interface aqui desenvolvida serve um objectivo específico, interagir com um sistema domótico, mais propriamente com o B-LIVE.

No desenvolvimento dos reconhecedores de fala foram alcançados dois objectivos distintos. O primeiro foi adquirir conhecimentos que permitiram construir um reconhecedor de fala de raiz (utilizando o HTK). O segundo foi demonstrar capacidade para integrar tecnologia de diferentes instituições, no sentido de desenvolver uma solução viável.

Também a realização de ensaios reais era um objectivo importante. **A partir das experiências realizadas, podemos dizer que as interfaces com reconhecimento de fala começam a ser uma realidade viável para diversas aplicações**. No caso específico das pessoas com limitações funcionais, estas interfaces representam uma nova janela de oportunidades.

Estas interfaces permitem maior autonomia e liberdade de movimentação. No caso concreto deste trabalho, a interface desenvolvida permitiu que pessoas que anteriormente estavam dependentes de um prestador de cuidados, por exemplo, para se movimentarem dentro de casa o pudessem fazer sem necessitarem de ajuda.

6.3 Sugestões para Continuação

As interface com reconhecimento de fala podem ser utilizadas para inúmeras aplicações, contudo nestas sugestões apenas vamos propor melhoramentos a este trabalho.

A versão actual da interface funciona em contínuo, isto é, quando ligada está sempre a fazer reconhecimento. Numa aplicação real isto não é viável, os erros no reconhecimento devido a conversas e ruído de fundo são uma constante. Uma forma de eliminar estes erros seria a introdução da técnica ***Keyword Spotting*** [49], para activar e desactivar a interface através da detecção de uma palavra chave.

A **interacção bidireccional com o utilizador** é também uma funcionalidade interessante. A interface passaria a ter capacidade para interrogar o utilizador sobre as opções que este toma, ou até de sugerir a execução de tarefas complementares à que foi ordenada pelo utilizador. Por exemplo, se o utilizador der ordem para abrir o estore do quarto, caso a luz esteja ligada a interface poderia sugerir ao utilizador para a desligar.

A integração da interface com um **sistema de localização**, poderá permitir maior controlo sobre as ordens dadas pelo utilizador. Por exemplo, poderíamos definir que as acções potencialmente perigosas (abertura de portas ou ligação de tomadas, por exemplo) apenas pudessem ser executadas se o utilizador estiver no local.

Para que o utilizador possa utilizar a interface mais comodamente seria interessante utilizar ***arrays*** **de microfones** para captar o sinal de áudio. A ideia seria colocar os microfones em pontos chave da casa de forma que a sua utilização pelo utilizador seja feita de uma forma transparente [12].

Tendo em conta as sugestões apresentadas pelos utilizadores que testaram a interface, de futuro deve ser estudada a possibilidade de integrar no sistema B-LIVE os electrodomésticos que é comum encontrar numa casa. Por exemplo, frigorífico, micro-ondas, maquinas de lavar/secar louça e roupa, vídeo porteiro, telefone, televisão e leitor de DVD. Depois de integrar estes electrodomésticos no B-LIVE, seria necessário actualizar a interface com reconhecimento de fala.

Anexo A

Alfabetos fonéticos

Um alfabeto fonético é um conjunto de símbolos criados para representar graficamente os sons da fala. A utilização de alfabetos fonéticos permite ter uma relação **biunívoca** entre um som e o seu símbolo. Pode-se dizer que cada som é representado apenas por um único símbolo.

Um dos alfabetos mais utilizados é o **Alfabeto Fonético Internacional** (*Internatoinal Phonetic Alphabet*) **(IPA)**, proposto pela Associação Internacional de Fonética [22] [50]. A figura A.1 apresenta os símbolos utilizados por este alfabeto para representar os diferentes sons da fala.

O alfabeto fonético **SAMPA** (*Speech Assessment Methods Phonetic Alphabet*), faz o mapeamento dos símbolos do IPA em ASCII, em particular, os de sete bits. Este alfabeto é muito útil quando se pretende fazer processamento em computador. A figura A.2, apresenta os símbolos utilizados pelo alfabeto fonético SAMPA.

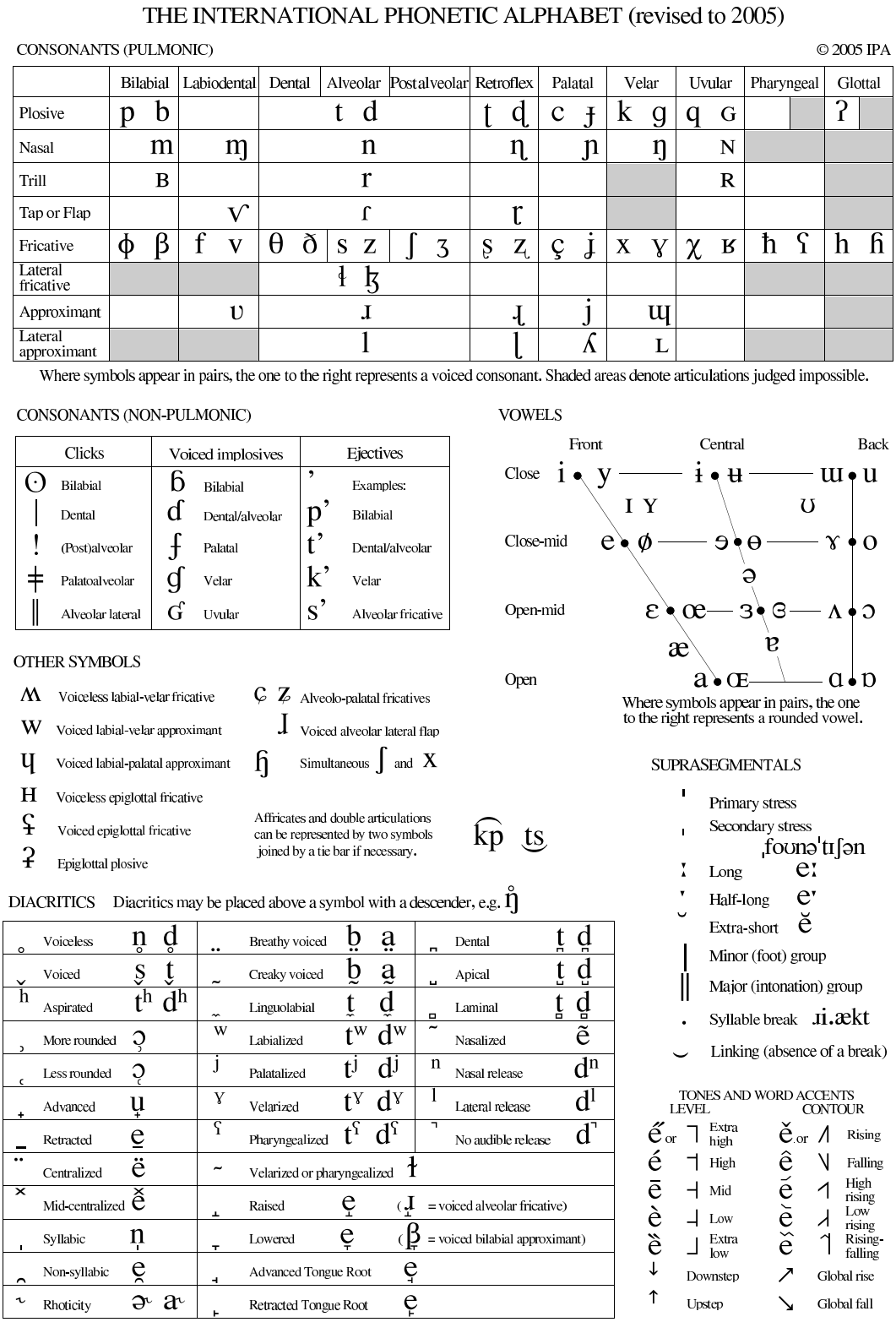


Figura A.1: Alfabeto fonético internacional [50].

Consoantes		
Oclusivas		
Símbolo	**Palavra**	**Transcrição**
p	pai	paj
b	barco	"barku
t	tenho	"teJu
d	doce	"dos@
k	com	ko~
g	grande	"gr6nd@
Fricativas		
f	falo	"falu
v	verde	"verd@
s	céu	sEw
z	casa	"kaz6
S	chapéu	S6"pEw
Z	jóia	"ZOj6
Nasais		
m	mar	mar
n	nada	"nad6
J	vinho	"viJu
Líquidas		
l	lanche	"l6nS@
L	trabalho	tr6"baLu
r	caro	"karu
R	rua	"Ru6

Vogais e ditongos				
Símbolo	**Palavra**	**Transcrição**	**Palavra**	**Transcrição**
i	vinte	"vint@	lápis	"lapiS
e	fazer	f6"zer		
E	belo	"bElu		
a	falo	"falu		
6	cama	"k6m6	madeira	m6"d6jr6
O	ontem	"Ont6~j~		
o	lobo	"lobu		
u	jus	ZuS	futuro	fu"turu
@	felizes	f@"liz@S		
i~	fim	fi~		
e~	emprego	e~"pregu (ou		
6~	irmã	ir"m6~		
o~	bom	bo~		
u~	um	u~		
aw	mau	maw etc.: iw,		
aj	mais	majS etc.: ej,		
6~j~	têm	t6~j~ etc.: e~		

Figura A.2: Alfabeto fonético SAMPA [23].

Anexo B

Reconhecedor de fala dependente do orador, baseado em HTK

Pelas razões enunciadas na secção 4.5, **vamos utilizar o *HTK Toolkit* para construir um reconhecedor de fala dependente do orador**. A metodologia que seguimos na sua construção é a apresentada no capítulo terceiro do livro ***The HTK Book***[44], a qual contém as seguintes fases:

1. Preparação dos dados
2. Criação dos modelos monofones
3. Criação dos modelos trifones
4. Avaliação
5. Reconhecimento em tempo real

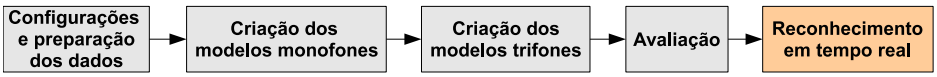


Figura B.1: Construção do reconhecedor.

Estas fases serão apresentadas detalhadamente ao longo desta secção.

B.1 Configurações e preparação dos dados

Para construir um reconhecedor de fala são necessários dados acústicos, tanto para treinar como para testar o reconhecedor [44]. A recolha dos dados acústicos é a primeira tarefa a realizar na construção de um reconhecedor de fala. Para tal é necessário percorrer os seguintes passos:

1. Definir uma gramática

2. Gerar e gravar os conjuntos de frases de treino e teste
3. Construir o dicionário
4. Criar os ficheiros com a transcrição fonética
5. Extrair os *feature vectors*

B.1.1 Gramática

Antes de mais é necessário definir o **cenário** onde o reconhecedor vai ser utilizado. Neste caso em concreto, o cenário é uma casa de habitação, onde o que se pretende é executar acções sobre os **diversos dispositivos**, isto é, controlar o **ambiente envolvente**. A figura B.2, apresenta o cenário típico.

Figura B.2: Cenário de utilização do reconhecedor.

No cenário apresentado pela figura B.2 são vários os dispositivos que pretendemos controlar. Em concreto, podemos enumerar os seguintes: **portas, lâmpadas, tomadas, estores e o autoclismo**. A acções que se podem executar sobre estes dispositivos são: **abrir/fechar, ligar/desligar e subir/descer**. Assim sendo, o conjunto de frases que se pretende reconhecer é o seguinte:

- Luz da Sala
- Luz da Sala dois
- Luz da Cozinha
- Luz da Casa de Banho

- Luz do Corredor
- Luz do Quarto
- Abrir a Porta da Frente
- Abrir a Porta do Quarto
- Abrir o Estore do Quarto
- Fechar a Porta da Frente
- Fechar a Porta do Quarto
- Fechar o Estore do Quarto
- Subir o Estore
- Descer o Estore
- Ligar todas as lâmpadas
- Ligar a Tomada do Quarto
- Desligar todas as lâmpadas
- Desligar a Tomada do Quarto
- Tomada do Quarto
- Autoclismo

No conjunto de frases sugerido existem algumas que não indicam qual a acção a executar. Isto acontece porque nestes casos o que se pretende é executar a acção que **inverta o estado actual** (lâmpadas), ou então porque só é possível executar **uma única acção** e neste caso está implícita (autoclismo).

A forma de definir estas frases é através de uma gramática. O *HTK* disponibiliza uma linguagem para definir formalmente uma gramática. A gramática da figura B.3, define formalmente as frases enumeradas. **Os parêntesis rectos delimitam palavras opcionais e as barras verticais separam as diversas possibilidades**.

De forma a melhorar a compreensão da gramática, esta pode ser representada graficamente. A figura B.4 apresenta todas as possibilidades permitidas pela gramática, para construir frases iniciadas pelas palavras ”Ligar” e ”Desligar”.

A forma como a gramática é apresentada na figura B.3 é utilizada apenas por conveniência, e é uma representação de alto nível, fácil de utilizar. O *HTK* utiliza a informação contida na gramática recorrendo a uma representação de baixo nível. Actualmente, **é usada uma rede onde estão representadas todas as palavras bem como a forma como estas se ligam entre si**. Esta notação de baixo nível chama-se ***HTK Standard Lattice Format* (SLF)**. O *HTK* disponibiliza uma ferramenta, o ***HParse***, que constrói a rede de palavras a partir da representação de alto nível da gramática [44].

```
$autoclismo = AUTOCLISMO;
$tomada = TOMADA;
$locais_com_tomada = QUARTO;
$subir_descer = SUBIR | DESCER;
$objectos_sobem_descem = ESTORE;
$dispensar = DISPENSAR;
$produtos_dispensar = PRODUTO;
$numero = UM | DOIS | TRES | QUATRO;
$ligar_desligar = LIGAR | DESLIGAR;
$luz = LUZ;
$locais_com_luz = CASA DE BANHO | COZINHA | QUARTO | CORREDOR;
$abrir_fechar = ABRIR | FECHAR;
$objectos_abrem_fecham = PORTA | ESTORE;
$locais_objectos_abrem_fecham = FRENTE | QUARTO;
(
   SENT-START
   (
      ($autoclismo) |
      ($tomada [DO] $locais_com_tomada) |
      ($subir_descer [O] $objectos_sobem_descem) |
      ($dispensar [O] $produtos_dispensar $numero) |
      ($ligar_desligar TODAS [AS] LAMPADAS) |
      ($ligar_desligar [A] TOMADA [DO] QUARTO) |
      ($luz [DA] SALA $numero) |
      ($luz [DA | DO] $locais_com_luz) |
      ($abrir_fechar DISPENSADOR) |
      ($abrir_fechar [A | O] $objectos_abrem_fecham [DA | DO] $locais_objectos_abrem_fecham)
   )
   SENT-STOP
)
```

Figura B.3: Gramática.

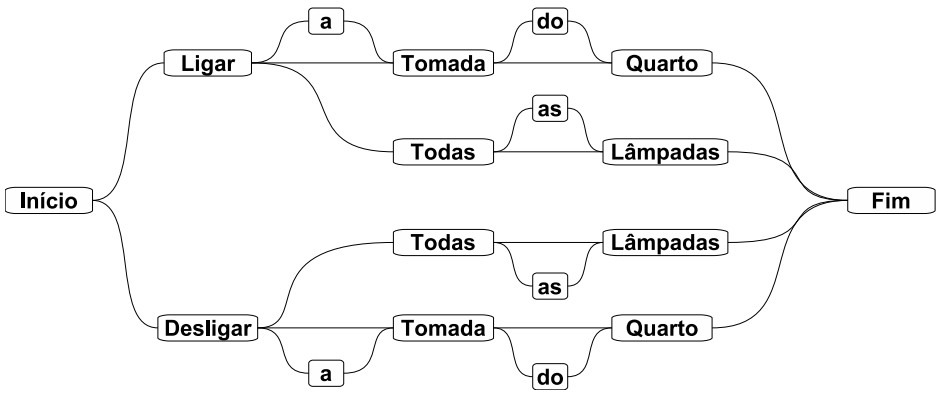


Figura B.4: Frases iniciadas por “Ligar”e “Desligar”.

Partindo do principio que a gramática está num ficheiro de nome ***gram***, a rede de palavras é criada executando o seguinte comando:

HParse gram wdnet,

onde *wdnet* é o nome do ficheiro que contém a rede de palavras.

B.1.2 Criação dos conjuntos de frases para treino e teste

Os conjuntos de frases para treino e teste do reconhecedor podem ser obtidos a partir da gramática presente no ficheiro *gram*. Utilizando a ferramenta ***HSGen*** disponibilizada pelo HTK, juntamente com um **dicionário genérico para o Português**, presente no ficheiro ***dicionario***. Basta executar os seguintes comandos para obter os **conjuntos de treino e de teste**:

HSGen -l -n 500 wdnet dicionario > TrainPrompts.txt

HSGen -l -n 100 wdnet dicionario > TestPrompts.txt

onde os ficheiros ***TrainPrompts.txt*** **e** ***TestPrompts.txt*** contêm os conjuntos de treino e de teste, respectivamente. Neste caso foram geradas **500 frases de treino e 100 de teste**. A figura B.5 apresenta as primeiras frases do conjunto de treino.

```
1. FECHAR ESTORE DA QUARTO
2. FECHAR O ESTORE FRENTE
3. FECHAR DISPENSADOR
4. AUTOCLISMO
5. DESCER O ESTORE
6. DESLIGAR A TOMADA QUARTO
7. DESLIGAR A TOMADA DO QUARTO
8. SUBIR ESTORE
9. ABRIR DISPENSADOR
10. ABRIR O PORTA FRENTE
11. DISPENSAR PRODUTO UM
12. ABRIR DISPENSADOR
13. SUBIR O ESTORE
14. SUBIR ESTORE
        ...
```

Figura B.5: Conjunto de treino.

B.1.3 Dicionário

O primeiro passo na construção de um dicionário é identificar todas as palavras presentes nas frases que se pretende reconhecer. O dicionário tem que conter a **representação fonética** de todas as **palavras que constituem estas frases**. A melhor forma de obter esta lista de palavras é extraí-la automaticamente do conjunto de frases de treino. O *script Prompts2Wlist.pl* fornecido

pela ferramenta *HTK*, permite efectuar esta tarefa de uma forma automática. Os parâmetros de entrada são o ficheiro que contém o conjunto de frases para treino e o ficheiro de saída que irá conter todas as palavras utilizadas nas frases de treino.

perl Prompts2Wlist.pl TrainPrompts.txt wlist,

onde *wlist* é o ficheiro de saída.

O dicionário contendo a pronúncia das palavras utilizadas pelo reconhecedor, pode ser obtido a partir da lista de palavras *wlist* e de um dicionário genérico. O *HTK* fornece uma aplicação, o ***HDMan***, para executar automaticamente esta tarefa. A utilização desta aplicação é feita da seguinte forma:

HDMan -m -w wlist -n monophones1 -l dlog dict dicionario,

onde ***dict*** é o dicionário a utilizar pelo reconhecedor. A opção -l permite criar um ficheiro de *log* com informação estatística sobre o dicionário, a opção -n permite criar uma lista com todos os fonemas utilizados pelo dicionário *dict*. Esta lista é guardada no ficheiro *monophones1*, ficheiro este que inclui também o modelo de silêncio *sil* como se pode ver na figura B.7. A figura B.6 apresenta o dicionário *dict*.

O dicionário tem a seguinte estrutura:

WORD [outsyn] p1 p2 p3 ...,

o que significa que a palavra *WORD* é pronunciada recorrendo à sequência de fonemas *p1 p2 p3 ...*. O conteúdo dos parêntesis rectos especifica o *output* do reconhecedor sempre que a palavra é reconhecida. Quando não é especificado nenhum *output* o reconhecedor retorna a própria palavra. Nos casos em que os parêntesis rectos não têm conteúdo, o reconhecedor não retorna nenhum resultado. É o que acontece no dicionário da figura B.6 para as entradas ***SENT-START*** **e** ***SENT-STOP*** que correspondem ao início e fim das frases, respectivamente, e cuja pronúncia corresponde ao **modelo de silêncio**.

B.1.4 Gravação das frases para treino e teste

Neste ponto da construção do reconhecedor é necessário **gravar o material de áudio correspondente às frases de treino e de teste**. A gravação das frases de treino e teste é feita recorrendo à ferramenta ***HSLab*** (figura B.8) fornecida pelo *HTK*.

Para que todo o processo seja feito de uma forma automática é necessário construir um *script* que execute os seguintes comandos para cada uma das frases do conjunto de treino e de teste:

HSLab train

copy train_0 S$num.wav,

onde ***train_0*** é o ficheiro gerado pelo comando *HSLab train* e *$num* é o número da frase no respec-

```
A               ax sp
ABRIR           ax b r ih r sp
AS              ax S sp
AUTOCLISMO      aw t O k l ih Z m u sp
BANHO           b ax J u sp
CASA            k a Z ax sp
CORREDOR        k u R d o r sp
COZINHA         k u Z i J ax sp
DA              d ax sp
DE              d sp
DESCER          d S s e r sp
DESLIGAR        d s l ih g a r sp
DISPENSADOR     d ih S p en s ax d o r sp
DISPENSAR       d ih S p en s a r sp
DO              d u sp
DOIS            d oj S sp
ESTORE          S t O r sp
FECHAR          f S a r sp
FRENTE          f r en t sp
LAMPADAS        l an p ax d ax S sp
LIGAR           l ih g a r sp
LUZ             l u S sp
O               u sp
PORTA           p O r t ax sp
PRODUTO         p r u du tu sp
QUARTO          ku a r t u sp
QUATRO          ku a t r u sp
SALA            s a l ax sp
SENT-START  []  sil
SENT-STOP   []  sil
SUBIR           s u b ih r sp
TODAS           t O d ax S sp
TOMADA          t u m a d ax sp
TRES            t r e S sp
UM              un sp
```

Figura B.6: Dicionário.

```
ax, sp, b, r, ih, S, aw, t, O, k, l, Z, m, u, J, a, R, d,
o, i, s, e, g, p, en, oj, f, an, du, tu, ku, un, sil
```

Figura B.7: Lista de fonemas utilizados.

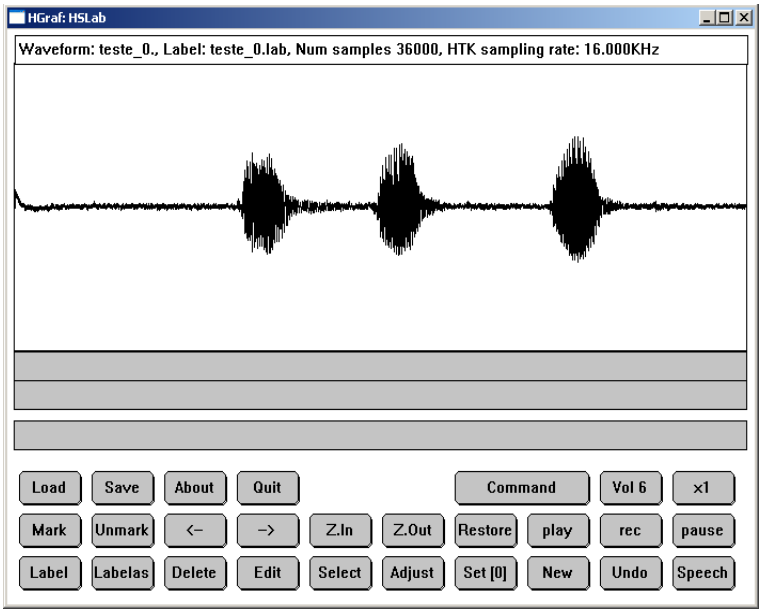


Figura B.8: Aplicação para gravação das frases de treino e teste.

tivo conjunto, neste caso no conjunto de treino. Desta forma, o material de áudio que corresponde à primeira e última frases do conjunto de treino será guardado nos ficheiros com os nomes ***S001.wav*** **e** ***S500.wav*** respectivamente.

Este procedimento tem que ser repetido para o conjunto de teste, tendo em conta as seguintes alterações:

HSLab test

copy test_0 T$num.wav,

assim os ficheiros com o material de áudio correspondente as frases de teste estão compreendidos entre ***T001.wav*** **e** ***T100.wav***.

B.1.5 Criação dos ficheiros com a transcrição fonética

Para que os dados que acabamos de obter sejam úteis é necessário gerar as respectivas transcrições fonéticas. Para o efeito, vamos utilizar as ferramentas disponibilizadas pelo *HTK*. O primeiro passo é criar um ***Master Label File*** **(MLF)** com as transcrições ao nível da palavra para cada um dos ficheiros de áudio. Para criar este ficheiro de uma forma automática podemos utilizar o ***script TrainPrompts2mlf.pl*** disponibilizado juntamente com o *HTK*. A sua utilização é a seguinte:

perl TrainPrompts2mlf.pl TrainWords.mlf TrainPrompts.txt

onde o parâmetro de entrada é o ficheiro com as frases de treino (TrainPrompts.txt) e o parâmetro de saída é o ficheiro MLF (TrainWords.mlf) contendo as transcrições ao nível da palavra. A figura

B.9 ilustra o conteúdo do ficheiro TrainWords.mlf.

```
#!MLF!#
"S001.lab"
FECHAR
ESTORE
DA
QUARTO
.
"S002.lab"
FECHAR
O
ESTORE
FRENTE
 ...
```

Figura B.9: Transcrições ao nível da palavra.

Agora que já possuímos as transcrições ao nível da palavra podemos avançar para a **transcrição fonética**. O ficheiro MLF contendo as transcrições fonéticas pode ser gerado de uma forma automática pela ferramenta ***HLEd***, por exemplo:

HLEd -d dict -i phones0.mlf mkphones0.led TrainWords.mlf

os parâmetros de entrada são **dict, mkphones0.led e TrainWords.mlf**, contendo o dicionário as opções que definem a forma como as transcrições irão ser geradas e as transcrições ao nível da palavra, respectivamente. O parâmetro de saída é o ficheiro MLF **phones0.mlf**, que contém as transcrições fonéticas. A figura B.11 apresenta um excerto do ficheiro phones0.mlf.

```
EX
IS sil sil
DE sp
```

Figura B.10: Configurações para gerar a transcrição fonética.

B.1.6 Extracção dos *feature vectors*

Esta é a última tarefa a realizar no que diz respeito à preparação dos dados. **Consiste na extracção de vectores com as características mais relevantes do sinal**, tendo em conta a tarefa que se pretende realizar. Neste caso concreto pretende-se extrair as características mais relevantes para reconhecimento de fala. Na literatura de língua Inglesa estes vectores têm o nome de ***feature vectors***.

A extracção destes vectores pode ser feita de uma forma automática utilizando a ferramenta ***HCopy***, fornecida pelo *HTK*. O comando utilizado foi o seguinte:

HCopy -T 1 -C config -S codetr.scp

```
#!MLF!#
"S001.lab"
sil
f
S
a
r
S
t
0
r
d
ax
ku
a
r
t
u
sil
.
"S002.lab"
sil
f
S
   ...
```

Figura B.11: Transcrições com monofones.

as opções utilizadas foram as seguintes: *-C config* esta opção indica que os parâmetros necessários para a criação dos *feature vectors* estão no ficheiro de configuração *config* e *-S codetr.scp* indica que o ficheiro *codetr.scp* contém uma lista com todos os ficheiros de dados e os correspondentes ficheiros de saída. Os ficheiros *config* e *codetr.scp* estão ilustrados nas figuras B.12 e B.13, respectivamente.

As configurações presentes no ficheiro *config* são as seguintes: TARGETKIND = MFCC_0_D_A, TARGETRATE = 100000.0 significa que o sinal acústico irá ser analisado tendo em conta frames de 10 ms, SAVECOMPRESSED = T indica que os dados resultantes desta operação devem ser guardados em formato comprimido, SAVEWITHCRC = T esta opção é necessária para que o *checksum* seja guardado juntamente com os dados de saída, WINDOWSIZE = 250000.0 define a largura da janela temporal a utilizar nos cálculos da FFT neste caso 25 ms, USEHAMMING = T indica que vai ser utilizada uma janela de Hamming, PREEMCOEF = 0.97 é co coeficiente de pré-enfasis , NUMCHANS = 26 será utilizado um banco de filtros com 26 canais, CEPLIFTER = 22 (Cepstral Liftering Coeficient), NUMCEPS = 12 significa que devem ser cálculados 12 coeficientes MFCC, ENORMALISE = F.

```
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = T
SAVEWITHCRC = T
WINDOWSIZE = 250000.0
USEHAMMING  = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = F
```

Figura B.12: Ficheiro de configuração.

```
S001.wav S001.mfc
S002.wav S002.mfc
S003.wav S003.mfc
S004.wav S004.mfc
S005.wav S005.mfc
S006.wav S006.mfc
S007.wav S007.mfc
S008.wav S008.mfc
S009.wav S009.mfc
S010.wav S010.mfc
```

Figura B.13: Correspondência entre os ficheiros de dados e de *feature vectors*.

B.2 Criação dos modelos monofones

Nesta fase da criação do nosso reconhecedor o que se pretende é obter um **conjunto de modelos monofones bem treinados**. Para tal, é necessário executar as seguintes tarefas: **inicialização dos modelos, ajuste dos modelo de silêncio, introdução do modelo para pausas curtas e realinhamento dos dados**. Em cada uma destas tarefas é necessário refinar os modelos. Isto é feito através da re-estimação dos mesmos.

B.2.1 Inicialização dos modelos monofones

O primeiro passo no sentido de treinar um conjunto de HMM, é criar o protótipo dos modelos. Neste primeiro passo o mais importante é definir o modelo e não os seus parâmetros. Nos sistemas que têm por unidade base o fonema é usual utilizar uma **topologia *left-right* com 3 estados** [44]. A figura B.14 ilustra a topologia referida, onde os vectores de médias e variâncias têm comprimento igual a 39, isto é 13 coeficientes MFCC mais 13 coeficientes delta mais 13 coeficientes de aceleração.

Depois de definir o protótipo para os HMM é necessário inicializá-lo. A inicialização do protótipo consiste em substituir o valor zero presente nos vectores de médias e o valor um dos

```
~o <VecSize> 39 <MFCC_0_D_A>
~h "proto"
<BeginHMM>
   <NumStates> 5
   <State> 2
   <Mean> 39
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
   <Variance> 39
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
   <State> 3
   <Mean> 39
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
   <Variance> 39
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
   <State> 4
   <Mean> 39
   0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 ...
   <Variance> 39
   1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 1.0 ...
<TransP> 5
0.0 1.0 0.0 0.0 0.0
0.0 0.6 0.4 0.0 0.0
0.0 0.0 0.6 0.4 0.0
0.0 0.0 0.0 0.7 0.3
0.0 0.0 0.0 0.0 0.0
<EndHMM>
```

Figura B.14: Protótipo para os HMMs.

vectores de variâncias pelos valores globais média e variância, respectivamente. Isto pode ser feito pela ferramenta ***HCompV***, bastando executar o seguinte comando:

HCompV -C config -f 0.01 -m -S Train.scp -M hmm0 proto

onde *config* é o ficheiro de configuração apresentado anteriormente, o ficheiro ***Train.scp*** contém uma lista com a localização dos ficheiros que contêm os *feature vectors*, por fim, o ficheiro ***proto*** contém o protótipo descrito anteriormente e apresentado na figura B.14.

Quanto às opções utilizadas, a opção *-f* faz com que seja criado um ficheiro macro de nome ***vFloors*** que contém o vector igual a 0.01 vezes a variância global. Este vector irá ser utilizado nos passos que se seguem indicando o limite mínimo para as variâncias que irão ser estimadas. A opção *-m* indica que as médias também devem ser calculadas.

Antes de iniciar o processo de criação dos HMM é necessário criar o ficheiro onde estes irão ser guardados. Em analogia ao que foi feito para as transcrições, também aqui se vai utilizar um *Master File* em concreto um ***Master Macro File* (MMF) de nome *hmmdefs***. O ficheiro *hmmdefs* deve conter uma cópia do protótipo para cada um dos fonemas presentes no ficheiro ***monophones0***.

Agora que já temos um ponto de partida para a criação dos modelos, **é necessário proceder à re-estimação dos mesmos até que sejam suficientemente robustos**. A ferramenta ***HERest***

permite efectuar esta tarefa. A sua utilização é a seguinte:

HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S Train.scp -H hmm0/macros -H hmm0/hmmdefs -M hmm1 monophones0

A ferramenta *HERest* carrega os modelos presentes em **hmm0** e procede à sua re-estimação utilizando os dados indicados no ficheiro *Train.scp*, os novos modelos são guardados em **hmm1**. A opção *-t* define o limite de ***pruning***. De forma a refinar os modelos é necessário executar o passo anterior mais duas vezes. O conjunto final de modelos será guardado em hmm3.

HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S Train.scp -H hmm1/macros -H hmm1/hmmdefs -M hmm2 monophones0

HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S Train.scp -H hmm2/macros -H hmm2/hmmdefs -M hmm3 monophones0

B.2.2 Ajuste do modelo de silêncio e introdução de pausas curtas

Nos passos anteriores foram gerados modelos com três estados para cada um dos fonemas em uso, bem como para o **modelo de silêncio *sil***. **O modelo de silêncio existente não permite a ocorrência de estados de silêncio consecutivos, pelo que é pouco robusto**. Para solucionar este problema vamos alterar o modelo de silêncio de forma que seja possível existir transições entre os estados dois e quatro em ambos os sentidos. As alterações ao modelo de silêncio estão definidas no ficheiro ***sil.hed***, apresentado na figura B.15. Os comandos AT presentes no ficheiro *sil.hed* têm a seguinte sintaxe:

AT i j prob itemList(t)

e o seu significado é o seguinte: Adicionar uma transição entre o estado *i* e *j* com probabilidade *prob* na matriz *t*. Neste caso concreto estamos a adicionar duas transições à matriz de transição de estados do modelo *sil*, uma entre o estado 2 e 4 e outra entre o estado 4 e 2 ambas com probabilidade 0.2.

```
AT 2 4 0.2 {sil.transP}
AT 4 2 0.2 {sil.transP}
AT 1 3 0.3 {sp.transP}
TI silst {sil.state[3], sp.state[2]}
```

Figura B.15: Comandos para ajustar o modelo de silêncio.

Vamos introduzir também o **modelo para pequenas pausas *sp***. O modelo *sp* tem apenas **um estado**, que corresponde ao estado central do modelo de silêncio. A figura B.16 apresenta o modelo *sp*. O ficheiro *sil.hed* contém comandos para o modelo *sp*, em concreto adição de uma transição entre o estado 1 e 3 e também uma ligação entre o estado 3 do modelo *sil* com o estado 2 do modelo *sp*.

O modelo *sp* foi introduzido no ficheiro *hmmdefs* presente em hmm3 e guardado em

```
~h "sp"
<BEGINHMM>
<NUMSTATES> 3
<STATE> 2
<MEAN> 39
 -1.266509e+001 1.676456e+000 5.713369e+000 2.722551e+000 ...
<VARIANCE> 39
 1.133274e+001 8.160398e+000 5.545747e+000 8.124187e+000 ...
<GCONST> 8.723096e+001
<TRANSP> 3
 0.0 0.5 0.5
 0.0 0.5 0.5
 0.0 0.0 0.0
<ENDHMM>
```

Figura B.16: Modelo para as pausas curtas.

hmm4. As alterações presentes no ficheiro *sil.led* são aplicadas pelo editor *HHEd*, bastando executar o seguinte comando:

HHEd -H hmm4/macros -H hmm4/hmmdefs -M hmm5 sil.hed monophones1

Como anteriormente, vamos refinar os modelos. Para isso é necessário gerar mais dois conjuntos de modelos **hmm6 e hmm7**. **Entre estes dois passos é necessário adicionar o modelo de silêncio ao dicionário *dict***.

HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S Train.scp -H hmm5/macros -H hmm5/hmmdefs -M hmm6 monophones1

HERest -C config -I phones0.mlf -t 250.0 150.0 1000.0 -S Train.scp -H hmm6/macros -H hmm6/hmmdefs -M hmm7 monophones1

B.2.3 Realinhamento dos dados

Os modelos existentes em hmm7 foram estimados prevendo a existência de pequenas pausas entre as palavras que constituem as frases que se pretende reconhecer, o que é uma boa aproximação à realidade. Contudo, as **transcrições existentes foram geradas partindo do principio que as palavras estavas encostadas umas às outras**, ou seja, não foi prevista a existência de pequenas pausas entre as palavras que constituem as frases. Para solucionar este problema **temos que introduzir as pequenas pausas existentes nos modelos acústicos na transcrição fonética**. Isto é feito com o seguinte comando:

HVite -o SWT -b silence -C config -a -H hmm7/macros -H hmm7/hmmdefs -i aligned.mlf -m -t 250.0 -y lab -I TrainWords.mlf -S Train.scp dict monophones1

Este comando gera novas transcrições fonéticas, tendo em conta as pausas entre as palavras. A figura B.17 apresenta um excerto do ficheiro *aligned.mlf* onde as alterações estão assinaladas.

```
#!MLF!#
"S001.lab"
sil
f
S
a
r
sp *
S
t
0
r
sp *
d
ax
sp *
ku
a
r
t
u
sp *
sil
  ...
```

Figura B.17: Introdução das pausas curtas.

Como anteriormente, é necessário estimar mais dois conjuntos de modelos, **hmm8 e hmm9**.

HERest -C config -I aligned.mlf -t 250.0 150.0 1000.0 -S Train.scp -H hmm7/macros -H hmm7/hmmdefs -M hmm8 monophones1

HERest -C config -I aligned.mlf -t 250.0 150.0 1000.0 -S Train.scp -H hmm8/macros -H hmm8/hmmdefs -M hmm9 monophones1

B.3 Criação dos modelos trifones

Neste passo o que se pretende é construir modelos dependentes do contexto, neste caso trifones. Os modelos dependentes do contexto contêm informação àcerca dos **fonemas vizinhos**, pelo que apresentam melhores resultados no reconhecimento. **Os modelos trifones são criados a partir dos modelos monofones existentes**.

B.3.1 Criação dos modelos trifones a partir dos monofones

Antes de mais é necessário criar novas transcrições fonéticas, substituindo os monofones pelos trifones correspondentes. Isto pode ser feito com a ferramenta ***HLEd***, executando o seguinte comando:

HLEd -n triphones1 -i wintri.mlf mktri.led aligned.mlf

A nova transcrição fonética é criada e guardada no ficheiro ***wintri.mlf*** apresentado em parte na figura B.18. Ao mesmo tempo é criada uma **lista com os trifones em utilização**. Esta lista é guardada no ficheiro ***triphones1*** e apresentada na figura B.19. O ficheiro ***mktri.led*** (figura B.20) contém os comandos utilizados pelo editor *HLEd* para gerar os trifones. O comando WB indica que não devem ser gerados trifones para os fonemas das extremidades e o comando TC indica que todos os monofones devem ser convertidos para trifones.

```
#!MLF!#
"S001.lab"
sil
f+S
f-S+a
S-a+r
a-r
sp
S+t
S-t+O
t-O+r
O-r
sp
d+ax
d-ax
sp
ku+a
ku-a+r
a-r+t
r-t+u
t-u
sp
sil
  ...
```

Figura B.18: Transcrições com trifones.

Agora é necessário criar novos modelos tendo em conta as transcrições com trifones. Estes podem ser gerados automaticamente a partir dos que já existem em **hmm9**, bastando executar o seguinte comando:

HHEd -B -H hmm9/macros -H hmm9/hmmdefs -M hmm10 mktri.hed monophones1

Como anteriormente, é necessário estimar mais dois conjuntos de modelos, **hmm11 e hmm12**.

HERest -B -C config -I wintri.mlf -t 250.0 150.0 1000.0 -s stats -S Train.scp -H hmm10/macros -H hmm10/hmmdefs -M hmm11 triphones1

HERest -B -C config -I wintri.mlf -t 250.0 150.0 1000.0 -s stats -S Train.scp -H hmm11/macros -H hmm11/hmmdefs -M hmm12 triphones1

```
sil
f+S
f-S+a
S-a+r
a-r
sp
S+t
S-t+0
t-0+r
0-r
d+ax
d-ax
ku+a
 ...
```

Figura B.19: Lista de trifones.

```
WB sp
WB sil
TC
```

Figura B.20: Configuração utilizada para gerar os modelos trifones.

B.4 Avaliação do reconhecedor

A avaliação dos modelos é fundamental para verificar se podemos parar o processo de treino ou, pelo contrário, se devemos continuar. A avaliação irá ser feita tanto aos modelos monofones como aos trifones. A ferramenta disponibilizada pelo *HTK* para avaliar os modelos criados é o ***HResults***. Contudo, esta ferramenta utiliza métricas ligeiramente diferentes das apresentadas no capítulo sobre *Reconhecimento de fala*.

As métricas utilizadas pelo *HResults* são as seguintes: **percentagem de frases reconhecidas correctamente (FRASES_CORR), percentagem de palavras reconhecidas correctamente (PALAVRAS_CORR) e a precisão com que as palavras foram reconhecidas (PALAVRAS_ACC)**.

$$\%FRASES\_CORR = 100\frac{\#CORR}{\#TOTAL} \tag{B.1}$$

$$\%PALAVRAS\_CORR = 100\frac{\#TOTAL - E - S}{\#TOTAL} \tag{B.2}$$

$$\%PALAVRAS\_ACC = 100\frac{\#TOTAL - E - S - I}{\#TOTAL} \tag{B.3}$$

onde E, S, I e # TOTAL representam **Eliminações, Substituições, Inserções e Total de frases (na equação B.1) ou palavras**. A avaliação é feita recorrendo aos dados acústicos previamente gravados para efeitos de teste do reconhecedor de fala.

B.4.1 Avaliação dos monofones

Para avaliar os **modelos monofones** utilizou-se o seguinte comando:

HResults -I TestWords.mlf monophones1 recout_mono.mlf

B.4.2 Avaliação dos trifones

O comando utilizado para avaliar os **modelos trifones** foi o seguinte:

HResults -I TestWords.mlf triphones1 recout_tri.mlf

B.5 Reconhecimento em tempo real

A aplicação disponibilizada pelo *HTK* para fazer reconhecimento em tempo real é o *HVite*. O comando utilizado foi o seguinte:

HVite -H hmm12/macros -H hmm12/hmmdefs -C config2 -w wdnet -p 0.0 -s 5.0 dict triphones1

Anexo C

Avaliação e questionário

Para avaliar a interface de reconhecimento de fala, vamos confrontar os utilizadores de teste com dois cenários diferentes. No final os utilizadores vão responder ao questionário proposto em C.3.

Avaliação

Antes da avaliação propriamente dita, vamos fazer uma pequena demonstração onde se vão pronunciar a generalidade dos comandos que a interface consegue reconhecer. Assim, os utilizadores de teste ficam a conhecer as funcionalidades disponíveis.

Para executar as tarefas propostas em cada um dos cenários, apresentados nas secções C.1 e C.2, os utilizadores vão pronunciar algumas das frases da tabela C.1. Para cada um dos comandos regista-se se o resultado do reconhecimento foi correcto ou não.

C.1 Cenário 1: O utilizador encontra-se na sala e pretende ir à casa de banho.

Uma situação possível para este cenário é a seguinte, o utilizador da interface está na sala a ver televisão e sente necessidade de ir à casa de banho. Inicialmente o utilizador está às escuras na sala, a porta da casa de banho está fechada e a luz desligada.

Tarefas a realizar pelo utilizador:

1. Ligar a luz da sala.
2. Abrir a porta da casa de banho.
3. Ligar a luz da casa de banho.
4. Fechar a porta da casa de banho.

Frase	True	False
Luz da Sala		
Luz da Sala dois		
Luz da Cozinha		
Luz da Casa de Banho		
Luz do Corredor		
Luz do Quarto		
Abrir a Porta da Frente		
Abrir a Porta do Quarto		
Abrir a Porta da Casa de Banho		
Abrir o Estore do Quarto		
Abrir o dispensador		
Fechar a Porta da Frente		
Fechar a Porta do Quarto		
Fechar a Porta da Casa de Banho		
Fechar o Estore do Quarto		
Fechar o dispensador		
Dispensar o produto um		
Dispensar o produto dois		
Dispensar o produto três		
Dispensar o produto quatro		
Subir o Estore do Quarto		
Descer o Estore do Quarto		
Ligar todas as lâmpadas		
Ligar a Tomada do Quarto		
Desligar todas as lâmpadas		
Desligar a Tomada do Quarto		
Tomada do Quarto		
Autoclismo		

Tabela C.1: Conjunto de frases para avaliação da interface de reconhecimento de fala.

5. ...
6. Autoclismo.
7. Abrir a porta da casa de banho.
8. Desligar a luz da casa de banho.
9. Desligar a luz da sala.

C.2 Cenário 2: O utilizador está no quarto e quer sair para a rua.

Neste cenário vamos considerar a situação seguinte, o utilizador está no quarto a descansar e sente vontade de ir dar um passeio. As condições iniciais são as seguintes: a luz do quarto e o aquecedor

estão ligados, a porta da frente está fechada.

Tarefas a realizar pelo utilizador:

1. Desligar a tomada do quarto.
2. Desligar a luz do quarto ou desligar todas as lâmpadas.
3. Abrir a porta da frente.
4. ...
5. Fechar a porta da frente.

C.3 Questionário

Questionário

As questões que se seguem referem-se à interface com reconhecimento de fala desenvolvida no decorrer deste trabalho.

Como classifica esta interface quanto à sua utilidade?
- □ Não tem nenhuma utilidade.
- □ É pouco útil.
- □ Tem alguma utilidade.
- □ É bastante útil.
- □ É muitíssimo útil.

Como classifica esta interface quanto à facilidade de utilização?
- □ Não consegui utilizar.
- □ Tive algumas dificuldades em utilizar.
- □ É muito fácil de utilizar.

Porquê?

Qual é a sua opinião àcerca das funcionalidades disponibilizadas pela interface?
- □ Demasiado poucas.
- □ Poucas.
- □ Suficientes.
- □ Todas as que são essenciais.
- □ Tem funcionalidades a mais, torna-se confusa a sua utilização.

Quais as funcionalidades que gostaria de adicionar à interface?

Acha positivo utilizar a fala para interagir com a casa?

□ Sim.

□ Não.

Porquê?

Se pudesse utilizar esta interface no seu dia a dia, a sua vida ficaria:

□ Na mesma.

□ Mais simplificada.

□ Mais complicada.

Porquê?

Na sua opinião, quais são os aspectos negativos desta interface?

Na sua opinião, quais são os aspectos positivos desta interface?

Sugestões?

Dados do utilizador:

Idade? _____ Sexo? _____ Nível da lesão? _____

Tem problemas respiratórios? _____

C.4 Respostas ao questionário

C.4.1 Utilizador M

Como classifica esta interface quanto à sua utilidade?
- ■ É muitíssimo útil.

Como classifica esta interface quanto à facilidade de utilização?
- ■ É muito fácil de utilizar.

Qual é a sua opinião àcerca das funcionalidades disponibilizadas pela interface?
- ■ Todas as que são essenciais.

Quais as funcionalidades que gostaria de adicionar à interface?

```
Frigorífico e micro-ondas.
```

Acha positivo utilizar a fala para interagir com a casa?
- ■ Sim.

Porquê?

```
Facilidade de utilização.
```

Se pudesse utilizar esta interface no seu dia a dia, a sua vida ficaria:
- ■ Mais simplificada.

Porquê?

```
Maior autonomia.
```

Na sua opinião, quais são os aspectos negativos desta interface?

```
Nenhum.
```

Na sua opinião, quais são os aspectos positivos desta interface?

```
Mobilidade, possibilidade de utilizar vários equipamentos.
```

Sugestões?

```
Novas funcionalidades.
```

C.4.2 Utilizador F

Como classifica esta interface quanto à sua utilidade?

- ■ É bastante útil.

Como classifica esta interface quanto à facilidade de utilização?

- ■ É muito fácil de utilizar.

Porquê?

Facilidade de utilização.

Qual é a sua opinião àcerca das funcionalidades disponibilizadas pela interface?

- ■ Todas as que são essenciais.

Quais as funcionalidades que gostaria de adicionar à interface?

Electrodomésticos e acesso aos prédios (porteiro).

Acha positivo utilizar a fala para interagir com a casa?

- ■ Sim.

Porquê?

Facilidade de utilização,
Liberta as mãos para outras acções (cadeira eléctrica),
Na posição deitada é difícil utilizar as outras interfaces.

Se pudesse utilizar esta interface no seu dia a dia, a sua vida ficaria:

- ■ Mais simplificada.

Porquê?

Maior autonomia e mobilidade,
Poder operar a partir de qualquer lugar.

Na sua opinião, quais são os aspectos negativos desta interface?

Nenhum.

Na sua opinião, quais são os aspectos positivos desta interface?

Simplicidade.

Sugestões?

Nenhuma.

Anexo D

Termo de consentimento informado

PARA PARTICIPAÇÃO NOS ENSAIOS À INTERFACE COM RECONHECIMENTO DE FALA PARA APOIO A PESSOAS COM LIMITAÇÕES FUNCIONAIS

A realização destes testes tem como finalidade avaliar uma interface com reconhecimento de fala, desenvolvida para interagir com o sistema domótico B-LIVE. O objectivo é avaliar o desempenho da interface em cenários de utilização real. Para o efeito, os pacientes terão que controlar diversos dispositivos presentes na habitação através de comandos verbais.

Eu abaixo assinado, declaro que tomei conhecimento dos objectivos do trabalho de investigação intitulado ”INTERFACE COM RECONHECIMENTO DE FALA PARA APOIO A PESSOAS COM LIMITAÇÕES FUNCIONAIS”, realizado por Carlos Jorge Enes Capitão de Abreu, no âmbito do Mestrado em Engenharia Biomédica - Ramo Instrumentação, Sinal e Imagem Médica ministrado pela Universidade de Aveiro.

Acrescento que estou informado de que todos os dados recolhidos serão tratados de modo estritamente confidencial, aceitando, por isso, fazer parte deste grupo de teste. Após ter sido devidamente informado declaro que tomei conhecimento dos objectivos dos testes e que aceito de livre vontade participar nos mesmos.

_____

Aveiro,_____de_____de 2007

Referências e Bibliografia

[1] Deborah Snoonian. Control systems: smart buildings. *IEEE Spectr.*, 40(8):18–23, 2003.

[2] Bo Yang Li Jiang, Da-You Liu. Smart home research. In *Machine Learning and Cybernetics, 2004*, volume 2, pages 659–663. IEEE Press, 2004.

[3] J. A. Gutierrez. On the use of ieee 802.15.4 to enable wireless sensor networks in building automation. volume 3, pages 1865–1869 Vol.3, 2004.

[4] Barralon P. Ye J. Rialle V. Demongeot J. Noury N., Virone G. New trends in health smart homes. In *Enterprise Networking and Computing in Healthcare Industry, 2003*, pages 118–127. IEEE Press, 2003.

[5] Won-Chul Bang Stefanov D.H., Zeungnam Bien. *The smart house for older persons and persons with physical disabilities: structure, technology arrangements, and perspectives*, volume 12, pages 228–250. IEEE Press, 2004.

[6] Censos 2001. Resultados definitivos, Instituto Nacional de Estatística de Portugal, 2001.

[7] Censos 2001, análise de população com deficiência. Resultados provisórios, Instituto Nacional de Estatística de Portugal, 2001.

[8] Santos V. Mota A. Silva V. Sizenando M. Bartolomeu P., Fonseca J. Automating home appliances for elderly and impaired people: The b-live approach. 2007.

[9] Régis Privat, Nadine Vigouroux, and Philippe Truillet. Usability of vocal interaction to access interactive systems by the elderly. *A.M.S.E.*, supplément 2C-2002:107–118, 2003.

[10] M. Conn, N.; McTear. Speech technology: a solution for people with disabilities. *Speech and Language Processing for Disabled and Elderly People (Ref. No. 2000/025), IEE Seminar on*, pages 7/1–7/6, 2000.

[11] Petr Cerva and Jan Nouza. Design and development of voice controlled aids for motor-handicapped persons. August 2007.

[12] Akira Sasou and Hiroaki Kojima. Noise robust speech recognition for voice driven wheelchair. August 2007.

[13] Soo-Young Suk and Hiroaki Kojima. Voice activated powered wheelchair with non-voice rejection algorithm. August 2007.

[14] Eiichi Ito. Multi-modal interface with voice and head tracking for multiple home appliances.

[15] www.microsoft.com/portugal/mldc/default.mspx, Novembro 2007.

[16] Peter Denes and Elliot Pinson. *The Speech Chain: The Physics and Biology of Spoken Language*, chapter The Speech Chain. Worth Publishers, 1993.

[17] Victória Fromkin and Robert Rodman. *Introdução à Linguagem*. Livraria Almedina, 1993.

[18] Rod. R. Seeley, Trent D. Stephens, and Philip Tate. *Anatomia & Fisiologia, 6ed*. LUSOCIÊNCIA - Edições Técnicas e Cientificas, Lda., 2003.

[19] Maria Mateus. *Fonética, Fonologia e Morfologia do Português*. Universidade Aberta, 1990.

[20] Cunha, Celso Cintra, and Luís F. Lindley. *Nova Gramática do Português Contemporâneo*, chapter Fonética e Fonologia. Edições João Sá da Costa, 2000.

[21] Jacob, Francone, and Lossow. *Anatomia e Fisiologia Humana*. Guanabara.

[22] Isabel Faria, Emília Pedro, Inês Duarte, and Carlos Gouveia. *Introdução à Linguística Geral e Portuguesa*. Caminho.

[23] Speech assessment methods phonetic alphabet, Janeiro 2008.

[24] Amabis & Martho. *Conceitos de Biologia Vol.2*. Editora Moderna, 2003.

[25] John Makhoul and Richard Schwartz. State of the art in continuos speech recognition. *BBN Systems and Technologies, Cambridge, MA 02138*, 1995.

[26] Carlos Jorge da Conceição Teixeira. *Reconhecimento de Fala de Oradores Estrangeiros*. PhD thesis, Universidade Técnica de Lisboa, 1998.

[27] Elisabete Marques Ranchhod. *Tratamento das Línguas por Computador*. CAMINHO, 2001.

[28] Claudio Becchetti and Lucio Prina Ricotti. *Speech Recognition: Theory and C++ Implementation*. John Wiley & Sons, Inc., New York, NY, USA, 1999.

[29] Lawrence Rabiner and Biing-Hwang Juang. *Fundamentals of Speech Recognition*. Prentice Hall International, Inc, 1993.

[30] Daniel Jurafsky and James H. Martin. *Speech and Language Processing*. Prentice Hall, 2000.

[31] Ronald A.Cole. *Survey of the State of the Art in Human Language Technology*. National Science Foundation.

[32] Lawrence R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. pages 267–296, 1990.

[33] www.rxtx.org, Outubro 2007.

[34] rxtx.qbang.org/wiki, Outubro 2007.

[35] www.javalobby.org/java/forums/t53333.html, Outubro 2007.

[36] E. Guglielmelli, P. Dario, C. Laschi, R. Fontanelli, M. Susani, P. Verbeeck, and J. Gabus. Humans and technologies at home: from friendly appliances to robotic interfaces. In *IEEE Internacional Workshop on Robot and Human Communication*, 1996.

[37] G. Nussbaum and K. Miesenberger. SmartX open source - A User Interface for Smart Environments. In *From Smart Homes to Smart Care*, 2005.

[38] S. Renouard, D. Menga, G. Brisson, G. Chollet, and M. Mokhtari. Toward a document based model for human-environment interaction. In *From Smart Homes to Smart Care*, 2005.

[39] www.lifetool.at, Outubro 2007.

[40] htk.eng.cam.ac.uk, Outubro 2007.

[41] www.speech.cs.cmu.edu/sphinx/twiki/bin/view/sphinx4/webhome, Novembro 2007.

[42] cmusphinx.sourceforge.net/sphinx4, Outubro 2007.

[43] Paul Lamere, Philip Kwok, Evandro B. Gouvêa, Bhiksha Raj, Rita Stingh, William Walker, and Peter Wolf. The cmu sphinx-4 speech recognition system.

[44] Steve Young, Dan Kershaw, Julian Odell, Dave Ollason, Valtcho Valtchev, and Phil Woodland. *The HTK Book*. 2000.

[45] http://msdn2.microsoft.com/en-us/library/ms720151.aspx, Dezembro 2007.

[46] www.microsoft.com/portugal/mldc/betaprograms/winclientdesktop.mspx, Novembro 2007.

[47] www.sensoryinc.com, Outubro 2007.

[48] www.parrotoem.com, Outubro 2007.

[49] K. Knill. Speaker dependent keyword spotting for accessing stored speech, 1994.

[50] http://www2.arts.gla.ac.uk/ipa/index.html, Janeiro 2008.